

**RGB-D ALGILAYICILAR KULLANILARAK  
EŞ ZAMANLI KONUM BELİRLEME VE HARİTALAMA**

**SIMULTANEOUS LOCALIZATION AND MAPPING  
USING RGB-D SENSORS**

**OĞUZHAN GÜÇLÜ**

**DOÇ. DR. AHMET BURAK CAN**

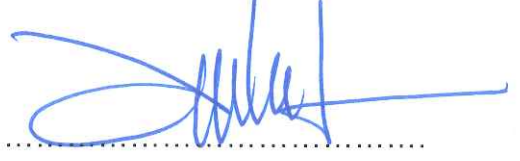
**Tez Danışmanı**

Hacettepe Üniversitesi  
Lisansüstü Eğitim - Öğretim ve Sınav Yönetmeliğinin  
Bilgisayar Mühendisliği Anabilim Dalı için Öngördüğü  
DOKTORA TEZİ olarak hazırlanmıştır.

2018

OĞUZHAN GÜÇLÜ'nün hazırladığı “RGB-D Algılayıcılar Kullanılarak Eş Zamanlı Konum Belirleme ve Haritalama” adlı bu çalışma aşağıdaki jüri tarafından BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI'nda DOKTORA TEZİ olarak kabul edilmiştir.

Prof. Dr. Mehmet Önder EFE  
Başkan



Doç. Dr. Ahmet Burak CAN  
Danışman



Doç. Dr. Selim AKSOY  
Üye



Dr. Öğr. Üyesi Adnan ÖZSOY  
Üye



Dr. Öğr. Üyesi Mehmet DEMİRCİ  
Üye



Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından DOKTORA TEZİ olarak onaylanmıştır.

Prof. Dr. Menemşe GÜMÜŞDERELİOĞLU  
Fen Bilimleri Enstitüsü Müdürü

## YAYINLAMA VE FİKRİ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezimin/raporumun tamamını veya herhangi bir kısmını, basılı (kâğıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma iznini Hacettepe Üniversitesi'ne verdiğimi bildiririm. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları bana ait olacaktır.

Tezin kendi orijinal çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

- Tezimin/Raporumun tamamı dünya çapında erişime açılabilir ve bir kısmı veya tamamının fotokopisi alınabilir.**  
(Bu seçenikle teziniz arama motorlarında indekslenebilecek, daha sonra tezinizin erişim statüsünün değiştirilmesini talep etmeniz ve kütüphane bu talebinizi yerine getirirse bile, tezinin arama motorlarının önbelleklerinde kalmaya devam edebilecektir.)
- Tezimin/Raporumun 15 / 05 / 2019 tarihine kadar erişime açılmasını ve fotokopi alınmasını (İç Kapak, Özet, İçindekiler ve Kaynakça hariç) istemiyorum.**  
(Bu sürenin sonunda uzatma için başvuruda bulunmadığım takdirde, tezimin/raporumun tamamı her yerden erişime açılabilir, kaynak gösterilmek şartıyla bir kısmı ve ya tamamının fotokopisi alınabilir)
- Tezimin/Raporumun ..... tarihine kadar erişime açılmasını istemiyorum, ancak kaynak gösterilmek şartıyla bir kısmı veya tamamının fotokopisinin alınmasını onaylıyorum.**
- Serbest Seçenek/Yazarın Seçimi**

25 / 05 / 2018

Oğuzhan GÜÇLÜ

*Aileme...*



## ETİK

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

15 / 05 / 2018



Oğuzhan GÜÇLÜ

## ÖZET

### **RGB-D ALGILAYICILAR KULLANILARAK EŞ ZAMANLI KONUM BELİRLEME VE HARİTALAMA**

**Oğuzhan GÜÇLÜ**

**Doktora, Bilgisayar Mühendisliği Bölümü**

**Tez Danışmanı: Doç. Dr. Ahmet Burak CAN**

**Mayıs 2018, 146 sayfa**

Otonom mobil bir robotun bilinmeyen bir ortamda görevini yerine getirebilmesi için, aynı anda hem ortamın haritasını oluşturması hem de harita üzerindeki anlık konumunu belirlemesi gereklidir. Literatürde Eş Zamanlı Konum Belirleme ve Haritalama (*Simultaneous Localization and Mapping*, SLAM) olarak adlandırılan bu problem, robotik alanındaki en temel araştırma alanlarından birisidir.

SLAM bağlamında, robotun ortamda gezinirken yapmış olduğu hareketlerin hesaplanması (odometri tahmini) ve ortamda daha önce bulunmuş olduğu bir alandan tekrar geçmekte olduğunun anlaşılması (döngü kapama tespiti) yetenekleri sistemin temelini oluşturur. Ortamın tutarlı bir haritasının oluşturulabilmesi için, robot hareketlerinin mümkün olduğunca düşük hata ile tahmin edilmesi ve döngü kapamaların başarılı şekilde tespit edilmesi gereklidir. Haritalanan ortam genişledikçe, bu iki temel işlevin etkili şekilde gerçekleştirilmesi daha da zorlaşmaktadır.

Bu tez kapsamında, RGB-D algılayıcıdan elde edilen renk ve derinlik çerçevelerinin birlikte kullanımıyla kapalı ortamların 3 boyutlu haritalarının

oluşturulması için çeşitli çalışmalar gerçekleştirilmiştir. İlk aşamada, bir analiz çalışması ile hareket tahmininin başarılı şekilde gerçekleştirilebilmesi için kullanılacak nitelik bulucu ve nitelik tanımlayıcı ikilisinin seçimi gerçekleştirilmiştir. Sonrasında, özellikle geniş ölçekli ortamlarda döngü kapamaların etkin şekilde tespit edilebilmesi için, genel ve yerel görüntü niteliklerinden faydalanan bir döngü kapama tespiti yöntemi geliştirilmiştir. Daha sonraki çalışmada ise, döngü kapama tespiti yöntemi daha hızlı ve verimli çalışması için genişletilmiştir. Bu amaçla, görüntü çerçevelerindeki yerel niteliklerin daha verimli şekilde kullanımına dayalı bir eşleşme yöntemi sisteme entegre edilmiş ve modele aykırı döngü kapama adaylarının dinamik şekilde elenmesini sağlayan bir mekanizma geliştirilmiştir. Son olarak ise, önceki çalışmalardan elde edilen birikimle, kapalı ortamların 3 boyutlu haritasını gerçek zamanlı çıkarabilen bir RGB-D SLAM sistemi geliştirilmiştir. Nitelik tabanlı hareket tahmini yapan bu sistem, genel görüntü niteliklerinin yaklaşık en yakın komşu aramasına imkân sağlayan bir ağaç yapısında indekslenmesi yoluyla döngü kapamaları tespit etmektedir.

Geliştirilen yöntemlere dair sonuçlar, bu alanda yaygın olarak kullanılan veri kümeleri üzerinde gerçekleştirilen kapsamlı deneyler neticesinde, literatürde standart haline gelmiş hata metriği kullanılarak analiz edilmiştir. İlk aşamalarda geliştirilen döngü kapama tespiti yöntemi, entegre edildiği RGB-D SLAM sisteminin başarımını önemli oranda arttırmış ve geniş ortamlarda çalışabilecek düzeye gelmesini sağlamıştır. Son aşamada geliştirilen SLAM sistemi ise, CPU üzerinde gerçek zamanlı olarak hem küçük hem de geniş ortamların haritalamasını etkin şekilde yapabilmekte ve diğer gelişmiş RGB-D tabanlı haritalama sistemlerinin çoğunluğunu performans açısından geride bırakmaktadır.

**Anahtar Kelimeler:** Eş Zamanlı Konum Belirleme ve Haritalama, Döngü Kapama Tespiti, Mekan Tanıma, Histogram, Autocorrelogram, Yerel Görüntü Nitelikleri

# **ABSTRACT**

## **SIMULTANEOUS LOCALIZATION AND MAPPING USING RGB-D SENSORS**

**Oğuzhan GÜÇLÜ**

**Doctor of Philosophy, Department of Computer Engineering**

**Supervisor: Assoc. Prof. Dr. Ahmet Burak CAN**

**May 2018, 146 pages**

In order to fulfill its mission in an unknown environment, an autonomous mobile robot needs to create map of the environment and locate itself instantly on the map at the same time. This problem, called Simultaneous Localization and Mapping (SLAM) in the literature, is one of the most fundamental research areas in the field of robotics.

In the context of SLAM, the abilities of computing the motions made by the robot while navigating in the environment (odometry estimation) and realizing that it is passing through an area again where it has previously visited (loop closure detection) form the basis of the system. In order to build a consistent map of the environment, the robot motions should be estimated with as low error as possible and loop closures should be detected successfully. As the mapping environment expands, performing these two basic functions effectively becomes even more difficult.

Within the scope of this thesis, various studies have been carried out to construct 3D maps of indoor environments by using color and depth frames together

obtained from an RGB-D sensor. In the first phase, the feature detector and feature descriptor pair that will be used for performing the motion estimation successfully has been chosen with an analysis study. After that, a loop closure detection method that exploits global and local image features has been developed to detect loop closures effectively, especially in large-scale environments. In the next study, the loop closure detection method has been extended to work faster and more efficiently. For this purpose, a matching method based on using local features in the image frames more efficiently has been integrated into the system and a mechanism that enables outlier loop closure candidates to be eliminated dynamically has been developed. Finally, an RGB-D SLAM system that is able to construct 3D maps of indoor environments in real time has been developed with the experience gained from previous studies. This system, which performs feature based motion estimation, detects loop closures through indexing global image features in a tree structure that enables approximate nearest neighbor search.

In consequence of extensive experiments carried out on widely used data sets in this area, the results regarding the developed methods have been analyzed using the error metric which has become standard in the literature, The loop closure detection method developed in the first phases has increased the performance of the RGB-D SLAM system that it was integrated considerably and made it capable of working in large environments. The SLAM system developed in the last stage can effectively map both small and large environments in real time on the CPU and outperforms most of the advanced RGB-D based mapping systems in performance.

**Keywords:** Simultaneous Localization and Mapping, Loop Closure Detection, Place Recognition, Histogram, Autocorrelogram, Local Image Features

## TEŞEKKÜR

Gerek lisans gerekse lisansüstü eğitim hayatımda çok büyük katkıları olan, doktora çalışmalarımı birlikte sürdürebilme fırsatını bana tanıyan, tez çalışması boyunca her alanda beni yönlendiren, her aşamada bilgi ve tecrübelerinden faydalanmamı sağlayan, bana olan güvenini ve desteğini hiçbir zaman esirgemeyen değerli hocam, tez danışmanım Sayın Doç. Dr. Ahmet Burak CAN'a,

Tez izleme aşamasında fikir ve önerileriyle tezin gelişmesine katkıda bulunan, değerli yorumlarıyla motivasyonumu arttıran tez izleme komitesi üyesi hocalarım Sayın Prof. Dr. Mehmet Önder EFE'ye ve Sayın Doç. Dr. Selim AKSOY'a,

Tez metnini inceleyerek içerik bakımından son halini almasına yardımcı olan, tez savunma sınavım sırasında önerileriyle bana katkıda bulunan Sayın Dr. Öğr. Üyesi Adnan ÖZSOY ve Sayın Dr. Öğr. Üyesi Mehmet DEMİRCİ'ye,

Bana mesleğimi sevdiren, engin bilgilerinden her fırsatta yararlandığım, desteğini benden eksik etmeyen değerli hocam Sayın Prof. Dr. Ali SAATÇİ'ye,

Çalışmalarım süresince bana her fırsatta yardımcı olan, büyük bir özveri ile bana yoldaşlık eden başta değerli dostlarım Ali ÇAĞLAYAN, Ali Osman SERHATOĞLU ve Onur MERTER olmak üzere tüm arkadaşlarıma,

Beni yetiştirip bu günlere getiren, her daim yanımda olan, benim için hiçbir fedakârlıktan kaçınmayan, sonsuz sevgi ve desteklerini bana sürekli hissettiren, bu hayattaki en büyük şansım olan aileme,

Canı gönülden teşekkür ederim.

# İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET .....	i
ABSTRACT .....	iii
TEŞEKKÜR .....	v
İÇİNDEKİLER .....	vi
ÇİZELGELER .....	ix
ŞEKİLLER .....	xii
SİMGELER VE KISALTMALAR .....	xvi
1. GİRİŞ .....	1
2. EŞ ZAMANLI KONUM BELİRLEME VE HARİTALAMA .....	6
2.1. SLAM Kapsamındaki Temel Zorluklar .....	8
2.1.1. Algılayıcı Ölçüm Belirsizlikleri .....	8
2.1.2. Veri İlişkilendirme Problemi .....	8
2.1.3. Ortam Büyüklüğü .....	8
2.1.4. Ortamın Dinamik Olması .....	9
2.2. SLAM Sistemlerinde Kullanılan Algılayıcılar .....	9
2.3. SLAM Probleminin Olasılıksal Tanımı .....	10
2.4. SLAM Problemi İçin Geliştirilen Temel Yöntemler .....	12
2.4.1. Kalman Filtresi Yöntemi .....	12
2.4.2. Parçacık Filtresi Yöntemi .....	15
2.4.3. Çizge Tabanlı Yöntem .....	18
3. İLGİLİ ÇALIŞMALAR .....	22
3.1. Nitelik Tabanlı Yöntemler .....	22
3.2. Yoğun Yöntemler .....	26
3.3. Diğer Yöntemler .....	30
3.4. Veri Kümeleri .....	31
3.4.1. ICL-NUIM Veri Kümesi .....	32
3.4.2. TUM RGB-D Benchmark Veri Kümesi .....	33
3.5. Ölçüm Metriği .....	34
4. RGB-D SLAM METOTLARINDAKİ NİTELİK BULUCU VE TANIMLAYICILARIN KARŞILAŞTIRILMASI .....	35
4.1. Kullanılan Nitelik Bulucu ve Tanımlayıcılar .....	35



4.2. RGB-D SLAM Sistemi .....	36
4.3. Deneysel Kurulum.....	38
4.4. Sonuçlar .....	40
4.5. Değerlendirme .....	44
5. GENEL VE YEREL GÖRÜNTÜ NİTELİKLERİ KULLANILARAK DÖNGÜ KAPAMA TESPİTİ.....	46
5.1. Döngü Kapama Tespiti Yönünden SLAM Yöntemlerinin Değerlendirilmesi ...	47
5.2. Döngü Kapama Tespiti Açısından RGB-D SLAM Sistemi.....	48
5.3. Önerilen Döngü Kapama Tespiti Yöntemi .....	49
5.3.1. Genel Tanımlayıcı Kullanılarak Mekân Tanıma .....	50
5.3.1.1. Histogram Karşılaştırması .....	51
5.3.1.2. Autocorrelogram Karşılaştırması .....	52
5.3.1.3. Döngü Kapama Adaylarının Belirlenmesi.....	55
5.3.2. Anahtar Nokta Eşleştirmesi ile Mekân Tanıma.....	56
5.4. Deneysel Kurulum.....	58
5.5. Sonuçlar .....	60
5.5.1. Histogram Benzerliği Deneyleri .....	60
5.5.2. Autocorrelogram Benzerliği Deneyleri .....	63
5.5.3. Histogram Benzerliği ile Nitelik Eşleştirmesi Deneyleri.....	66
5.5.4. Autocorrelogram Benzerliği ile Nitelik Eşleştirmesi Deneyleri.....	68
5.5.5. İşlemsel Performans.....	71
5.5.6. Tartışma .....	74
5.6. Değerlendirme .....	79
6. DÖNGÜ KAPAMA TESPİTİ YÖNTEMİNİN DAHA HIZLI VE ETKİLİ HALE GETİRİLMESİ.....	81
6.1. Önerilen Genişletilmiş Döngü Kapama Tespiti Yöntemi.....	81
6.1.1. Sıradüzensel Kümeleme ile Mekân Tanıma .....	82
6.1.2. Uyarlanabilir Eşikleme ile Mekân Tanıma.....	85
6.2. Deneysel Kurulum.....	87
6.3. Sonuçlar .....	88
6.3.1. Histogram Benzerliği ile Sıradüzensel Kümeleme Deneyleri.....	89
6.3.2. Autocorrelogram Benzerliği ile Sıradüzensel Kümeleme Deneyleri.....	91
6.3.3. Uyarlanabilir Eşikleme için Benzerlik Eşiğinin Belirlenmesi .....	94
6.3.4. Histogram Benzerliği ile Uyarlanabilir Eşikleme Deneyleri .....	98

6.3.5. Autocorrelogram Benzerliği ile Uyarlanabilir Eşikleme Deneyleri .....	101
6.3.6. İşlemsel Performans ve Tartışma .....	104
6.4. Değerlendirme .....	108
7. GENİŞ KAPALI ORTAMLAR İÇİN HIZLI RGB-D SLAM YAKLAŞIMI.....	110
7.1. Yöntem.....	110
7.1.1. <i>k</i> -SLAM Sistemine Genel Bakış .....	110
7.1.2. Nitelik Tabanlı Hareket Tahmini .....	111
7.1.3. Döngü Kapama Tespiti.....	112
7.1.3.1. Autocorrelogram Kullanımı .....	112
7.1.3.2. Anahtar Çerçeve Autocorrelogram Veritabanı.....	113
7.1.3.3. Uyarlanabilir Eşikleme .....	117
7.1.3.4. Robot Yolu Tahmini ve Harita Oluşturulması.....	118
7.2. Deneysel Kurulum.....	119
7.3. Sonuçlar.....	120
7.3.1. ICL-NUIM Veri Kümesi Deneyleri .....	120
7.3.2. TUM RGB-D Benchmark Veri Kümesi Deneyleri.....	122
7.3.2.1. <i>fr1</i> Görüntü Kayıtları .....	122
7.3.2.2. <i>fr2</i> Görüntü Kayıtları .....	124
7.3.3. İşlemsel Performans.....	129
7.4. Değerlendirme .....	132
8. SONUÇLAR .....	134
KAYNAKLAR.....	139
ÖZGEÇMİŞ .....	145

## ÇİZELGELER

### Sayfa

Çizelge 4.1. Deneyleerde kullanılan <i>fr1</i> görüntü kayıtları hakkında ayrıntılı bilgiler	39
Çizelge 4.2. Deneyleerde uygulanan parametre değerleri .....	39
Çizelge 4.3. <i>fr1</i> kayıtları için mutlak yol hatası (RMS-ATE) sonuçları (m) .....	40
Çizelge 4.4. <i>fr1</i> kayıtları için çerçeve başına işlem süresi (ms) .....	43
Çizelge 5.1. Deneyleerde kullanılan <i>fr2</i> görüntü kayıtları hakkında ayrıntılı bilgiler	58
Çizelge 5.2. Deneyleerde uygulanan parametre değerleri .....	59
Çizelge 5.3. RGB histogram benzerliği tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m) .....	60
Çizelge 5.4. Gri tonlamalı histogram benzerliği tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m) .....	61
Çizelge 5.5. RGB autocorrelogram benzerliği tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m) .....	63
Çizelge 5.6. Gri tonlamalı autocorrelogram benzerliği tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m) .....	64
Çizelge 5.7. RGB histogram benzerliği ve nitelik eşleştirme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m) .....	66
Çizelge 5.8. Gri tonlamalı histogram benzerliği ve nitelik eşleştirme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m) .....	67
Çizelge 5.9. RGB autocorrelogram benzerliği ve nitelik eşleştirme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m) .....	69
Çizelge 5.10. Gri tonlamalı autocorrelogram benzerliği ve nitelik eşleştirme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m) .....	69
Çizelge 5.11. Tüm uzaklık metrikleri için, önerilen teknikler ve RGB-D SLAM [44] ile elde edilen çerçeve başına işlem süreleri (ms) .....	72
Çizelge 6.1. Deneyleerde uygulanan parametre değerleri .....	88
Çizelge 6.2. RGB histogram benzerliği ve sıradüzensel kümeleme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m) .....	89

Çizelge 6.3. Gri tonlamalı histogram benzerliği ve sıradüzensel kümeleme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m).....	90
Çizelge 6.4. RGB autocorrelogram benzerliği ve sıradüzensel kümeleme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m).....	92
Çizelge 6.5. Gri tonlamalı autocorrelogram benzerliği ve sıradüzensel kümeleme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m) .....	92
Çizelge 6.6. Histogram kullanımında <i>eşikleme_faktörü</i> parametresi için belirlenen değerler .....	97
Çizelge 6.7. Autocorrelogram kullanımında <i>eşikleme_faktörü</i> parametresi için belirlenen değerler.....	97
Çizelge 6.8. RGB histogram benzerliği ve uyarlanabilir eşikleme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m).....	98
Çizelge 6.9. Gri tonlamalı histogram benzerliği ve uyarlanabilir eşikleme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m).....	99
Çizelge 6.10. RGB autocorrelogram benzerliği ve uyarlanabilir eşikleme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m) .....	101
Çizelge 6.11. Gri tonlamalı autocorrelogram benzerliği ve uyarlanabilir eşikleme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m) .....	102
Çizelge 6.12. Tüm uzaklık metrikleri için, sıradüzensel kümeleme ve uyarlanabilir eşikleme teknikleri ile RGB-D SLAM [44] için çerçeve başına işlem süreleri (ms)....	104
Çizelge 7.1. Deneylerde kullanılan ICL-NUIM görüntü kayıtları hakkında ayrıntılı bilgiler .....	120
Çizelge 7.2. ICL-NUIM veri kümesinde yer alan gürültüsüz kayıtlar üzerinde mutlak yol hatası (RMS-ATE) karşılaştırması (m) .....	121
Çizelge 7.3. ICL-NUIM veri kümesinde yer alan simüle edilmiş gürültü içeren kayıtlar üzerinde mutlak yol hatası (RMS-ATE) karşılaştırması (m) .....	121
Çizelge 7.4. TUM RGB-D Benchmark veri kümesinde yer alan <i>fr1</i> kayıtları üzerinde mutlak yol hatası (RMS-ATE) karşılaştırması (m).....	123

Çizelge 7.5. TUM RGB-D Benchmark veri kümesinde yer alan <i>fr2</i> kayıtları üzerinde mutlak yol hatası (RMS-ATE) karşılaştırması (m).....	125
Çizelge 7.6. TUM RGB-D Benchmark veri kümesinde yer alan <i>fr2</i> kayıtları üzerinde mutlak yol hatası (RMS-ATE) karşılaştırması-2 (m).....	125
Çizelge 7.7. TUM RGB-D Benchmark veri kümesinde yer alan <i>fr2</i> kayıtları üzerinde mutlak yol hatası (RMS-ATE) karşılaştırması-3 (m).....	125
Çizelge 7.8. Diğer nitelik tabanlı SLAM sistemleri ile işlemsel performans karşılaştırması (ms).....	132

## ŞEKİLLER

	<u>Sayfa</u>
Şekil 2.1. SLAM problemine dair örnek senaryo [9].....	7
Şekil 2.2. Topolojik ve geometrik harita örnekleri [10] .....	7
Şekil 2.3. SLAM probleminin grafiksel gösterimi [3].....	11
Şekil 2.4. Genişletilmiş Kalman Filtresi yönteminde kullanılan durum vektörü ( $\mu$ ) ve kovaryans matrisi ( $\Sigma$ ) [10]. .....	13
Şekil 2.5. FastSLAM algoritmasının yapısı (GKF: Genişletilmiş Kalman Filtresi) .	17
Şekil 2.6. Konum çizgesinin yapısı .....	19
Şekil 2.7. Çizge tabanlı SLAM yaklaşımı .....	20
Şekil 3.1. ICL-NUIM veri kümesinden çeşitli sahneler .....	32
Şekil 3.2. TUM RGB-D Benchmark veri kümesinden çeşitli sahneler.....	33
Şekil 3.3. Mutlak yol hatasının hesaplanması.....	34
Şekil 4.1. RGB-D SLAM sisteminin [44] genel işleyişi .....	37
Şekil 4.2. En başarılı 6 yönteme ait, <i>fr1</i> kayıtları için ayrıntılı hata oranları (Aday parametresinin 4, 8 ve 12 değerleri için elde edilen sonuçların ortalaması)... ..	41
Şekil 4.3. <i>fr1/desk</i> ve <i>fr1/plant</i> kayıtları için robot yoluna ait kesin referansın ve CenSurE+FREAK ile tahmin edilen sonucun x-y düzlemine izdüşümü (Aday = 8)... ..	42
Şekil 4.4. En başarılı 6 yönteme ait, <i>fr1</i> kayıtları için çerçeve başına işlem süreleri (Aday parametresinin 4, 8 ve 12 değerleri için elde edilen sonuçların ortalaması)... ..	44
Şekil 5.1. Kayma hatasının döngü kapama tespiti ile azaltılması .....	46
Şekil 5.2. Önerilen döngü kapama tespiti yönteminin RGB-D SLAM sistemi [44] ile entegrasyonu.....	49
Şekil 5.3. Önerilen döngü kapama tespiti yönteminin genel işleyişi.....	50
Şekil 5.4. Autocorrelogram hesaplama örneği.....	54
Şekil 5.5. Genel tanımlayıcı kullanılarak aday seçimi .....	55

Şekil 5.6. Anahtar nokta eşleştirmesi ile aday seçimi .....	57
Şekil 5.7. RGB histogram benzerliği tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması .....	62
Şekil 5.8. Gri tonlamalı histogram benzerliği tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması .....	62
Şekil 5.9. RGB autocorrelogram benzerliği tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması .....	65
Şekil 5.10. Gri tonlamalı autocorrelogram benzerliği tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması .....	65
Şekil 5.11. RGB histogram benzerliği ve nitelik eşleştirme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması .....	67
Şekil 5.12. Gri tonlamalı histogram benzerliği ve nitelik eşleştirme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması .....	68
Şekil 5.13. RGB autocorrelogram benzerliği ve nitelik eşleştirme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması .....	70
Şekil 5.14. Gri tonlamalı autocorrelogram benzerliği ve nitelik eşleştirme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması .....	71
Şekil 5.15. CenSurE nitelik bulucusu ile tespit edilen, her bir kayıt için çerçeve başına ortalama anahtar nokta sayıları .....	73
Şekil 5.16. <i>fr2/desk</i> ve <i>fr2/pioneer_360</i> kayıtları için anahtar nokta tespiti ve eşleşme örnekleri .....	74
Şekil 5.17. RGB-D SLAM [44] ve önerilen yöntem (en başarılı kombinasyon olan RGB+Relative+128+2) arasında doğruluk karşılaştırması .....	75
Şekil 5.18. RGB-D SLAM [44] ve en başarılı kombinasyon olan RGB+Relative+128+2 için detaylı mutlak yol hatası sonuçları .....	76
Şekil 5.19. En başarılı kombinasyon olan RGB+Relative+128+2 ile örnek kayıtlar için tahmin edilen yolun ve kesin referansın x-y düzlemine izdüşümü.....	77
Şekil 5.20. Örnek çerçeveler için, en başarılı kombinasyon olan RGB+ Relative+128+2 kullanılarak belirlenen döngü kapama adayları.....	78



Şekil 6.1. Önerilen genişletilmiş döngü kapama tespiti yönteminin genel görünümü .....	82
Şekil 6.2. Sıradüzensel kümeleme ağacı oluşturma .....	83
Şekil 6.3. Sıradüzensel kümeleme ağacında arama .....	84
Şekil 6.4. Sıradüzensel kümeleme ile aday seçimi .....	85
Şekil 6.5. Uyarlanabilir eşikleme mekanizması.....	86
Şekil 6.6. RGB histogram benzerliği ve sıradüzensel kümeleme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması.....	91
Şekil 6.7. Gri tonlamalı histogram benzerliği ve sıradüzensel kümeleme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması .....	91
Şekil 6.8. RGB autocorrelogram benzerliği ve sıradüzensel kümeleme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması .....	93
Şekil 6.9. Gri tonlamalı autocorrelogram benzerliği ve sıradüzensel kümeleme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması .....	94
Şekil 6.10. Aday sıralamasına göre, tüm görüntü kayıtları üzerinde ortalama histogram benzerliği dağılımı.....	95
Şekil 6.11. Aday sıralamasına göre, tüm görüntü kayıtları üzerinde ortalama autocorrelogram benzerliği dağılımı .....	95
Şekil 6.12. En benzer aday ile diğer adaylar arasındaki histogram benzerliği oranının değişimi .....	96
Şekil 6.13. En benzer aday ile diğer adaylar arasındaki autocorrelogram benzerliği oranının değişimi .....	96
Şekil 6.14. RGB histogram benzerliği ve uyarlanabilir eşikleme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması.....	100
Şekil 6.15. Gri tonlamalı histogram benzerliği ve uyarlanabilir eşikleme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması .....	100
Şekil 6.16. RGB autocorrelogram benzerliği ve uyarlanabilir eşikleme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması .....	103
Şekil 6.17. Gri tonlamalı autocorrelogram benzerliği ve uyarlanabilir eşikleme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması .....	103

Şekil 6.18. Geliştirilen mekân tanıma tekniklerinin işlemsel maliyet karşılaştırması..	105
Şekil 6.19. RGB-D SLAM [44] ve önerilen genişletilmiş yöntem (en etkili kombinasyon olan Gri+Relative+128+2) arasında doğruluk karşılaştırması.....	106
Şekil 6.20. RGB-D SLAM [44] ve en etkili kombinasyon olan Gri+Relative+128+2 için detaylı mutlak yol hatası sonuçları .....	107
Şekil 6.21. Genişletilmiş yöntemdeki en etkili kombinasyon olan Gri+Relative+128+2 ile örnek kayıtlar için tahmin edilen yolun ve kesin referansın x-y düzlemine izdüşümü .....	108
Şekil 7.1. <i>k</i> -SLAM sisteminin genel mimarisi.....	111
Şekil 7.2. Anahtar Çerçeve Autocorrelogram Veritabanı'nın yapısı .....	113
Şekil 7.3. Öncelik araması <i>k</i> -ortalamlar ağacını artırımsal olarak oluşturma ....	115
Şekil 7.4. Anahtar Çerçeve Autocorrelogram Veritabanı'nda döngü kapama adayı arama .....	116
Şekil 7.5. Uyarlanabilir eşikleme tekniği ile modele aykırı adayların elenmesine dair örnekler .....	118
Şekil 7.6. Örnek görüntü kayıtları için, tahmin edilen yol ve kesin referansın x-y düzlemine izdüşümü.....	126
Şekil 7.7. Her bir kayıt için elde edilen çerçeve başına ortalama anahtar nokta sayıları.....	127
Şekil 7.8. Örnek kayıtlar için sistem tarafından üretilen nokta bulutu gösterimleri ....	129
Şekil 7.9. Her bir kayıt için elde edilen çerçeve başına işlem süresi (ms) .....	130
Şekil 7.10. Her bir kayıt için elde edilen anahtar çerçeve sayısı.....	131

## SİMGELER VE KISALTMALAR

### Simgeler

$x$	Robot Konumu
$m$	Harita
$z$	Robot Gözlemi
$u$	Odometri Verileri
$\mu$	Durum Vektörü
$\Sigma$	Kovaryans Matrisi
$\Omega$	Bilgi Matrisi
$I$	Görüntü
$H$	Histogram
$\gamma$	Autocorrelogram

### Kısaltmalar

SLAM	Simultaneous Localization and Mapping
İHA	İnsansız Hava Aracı
3B	Üç Boyutlu
CPU	Central Processing Unit
MCL	Monte Carlo Localization
RANSAC	Random Sample Consensus
ICP	Iterative Closest Point
g2o	General Graph Optimization
SDF	Signed Distance Function
TSDF	Truncated Signed Distance Function
GPU	Graphics Processing Unit
ATE	Absolute Trajectory Error
RMS	Root Mean Square
GFTT	Good Features to Track

# 1. GİRİŞ

Mobil bir robotun otonom şekilde hareket ederek görevini yerine getirebilmesi için bulunduğu ortamdaki konumunu bilmesi gereklidir. Ortam haritasının mevcut olduğu durumlarda robot, sahip olduğu algılayıcılardan elde ettiği verilerle harita içerisinde karşılaştırma yaparak haritaya göre bulunduğu konumu hesaplayabilir. Robot bilinmeyen bir ortamda ise, eş zamanlı olarak hem ortamın haritasını çıkarması (ortamı öğrenmesi) hem de harita üzerinde anlık olarak konumunu belirlemesi gerekmektedir. Bu problem literatürde Eş Zamanlı Konum Belirleme ve Haritalama (*Simultaneous Localization and Mapping*, SLAM) olarak adlandırılmaktadır. SLAM, otonom robotların geliştirilmesi kapsamında yer alan en önemli problemlerden birisi olarak kabul edilmektedir [1].

SLAM, robotik alanındaki en aktif çalışma alanlarından birisidir. Özellikle son 30 yılda bu alana olan ilgi daha da artmıştır. Günümüzdeki teknolojik gelişmeler ışığında otonom robotların günlük hayata entegrasyonu gittikçe artmakta olup, SLAM konusu gün geçtikçe popülerliğini arttırmaktadır. SLAM geniş bir uygulama alanına sahiptir; kapalı ya da açık ortamlarda, yer altında, su altında, karada, havada ya da uzayda otonom olarak çalışan robotlar için temel gereksinimdir. Örneğin evlerde çalışan temizlik robotları, açık alanda kullanılabilen çim biçme makineleri, su altında çalışan ya da yer altında maden aramak için kullanılan keşif robotları, doğal afet durumlarında enkaz altında arama - kurtarma faaliyetleri için geliştirilen robotlar, gezegenler arası araştırmalar için kullanılan keşif robotları, özellikle günümüzde çok yaygın hale gelmiş olan insansız hava araçları, ve ulaşım alanında devrimsel bir yenilik olan otonom arabalar SLAM araştırmalarının popüler uygulama alanları arasındadır. SLAM, mobil robotun izleyeceği rotaya karar verebilmesi ve yapacağı yer değiştirmeyi dinamik şekilde planlayabilmesi için temel teşkil etmektedir [2].

Robotun ortam haritasını çıkarabilmesi için konumunu bilmesi gereklidir. Diğer taraftan, robotun konumunu hesaplayabilmesi için ortamın haritasına ihtiyacı vardır. Dolayısıyla bu iki problem eş zamanlı olarak çözülmeli; yani robot bir yandan algılayıcılardan gelen verileri yorumlayarak ortam haritasını oluşturabilmeli, diğer yandan ise haritanın mevcut hali üzerindeki göreceli konumunu hesaplayabilmelidir. Bu yönüyle SLAM, yumurta - tavuk problemine

benzemekte olup, robotik alanındaki en zor problemlerden birisi olarak değerlendirilmektedir [3].

Geliştirilen SLAM yönteminin tasarımı kullanılan algılayıcıya göre şekillenmektedir. Literatürde lazer ya da sonar mesafe algılayıcılarla çalışan yöntemler olduğu gibi, tek ya da çoklu kamera kullanımına dayalı görsel SLAM yaklaşımları da mevcuttur. Son dönemde ise RGB-D algılayıcıların piyasaya sürülmesi ile bu alandaki çalışmalar daha da popülerlik kazanmış olup, RGB-D algılayıcı tabanlı SLAM yöntemleri yaygınlaşmaya başlamıştır. Bu hususta en önemli etken, RGB-D algılayıcıların hem görsel hem de derinlik verisini bir arada gerçek zamanlı olarak sağlayabilmesidir.

Modern SLAM sistemlerinde haritalama problemi çizge tabanlı (*graph based*) yöntem [4] ile ele alınmaktadır. Bu yöntemde robotun ortamdaki gezinimi süresince bulunduğu konumlar çizgenin düğümlerinde, konumlar arasındaki uzamsal kısıtlar (*spatial constraint*) ise düğümleri birbirine bağlayan kenarlarda tutulmaktadır. Konum çizgesi adı verilen bu yapı kullanılarak robotun ortamda yapmış olduğu yolu (*trajectory*) oluşturan konumlar gerçeğe en yakın şekilde tahmin edilmeye çalışılmaktadır. Konumlar belirlendikten sonra ise, her bir konumda elde edilmiş olan algılayıcı verileri ortak bir koordinat sisteminde birleştirilerek ortam haritası oluşturulmaktadır.

Çizge tabanlı yöntemde konum çizgesi, odometri tahmini (*odometry estimation*) ve döngü kapama tespiti (*loop closure detection*) bileşenleri tarafından oluşturulur. Algılayıcıdan art arda gelen veri çerçeveleri (*frame*) karşılaştırılarak (*registration*) robotun gerçekleştirdiği süregelen hareketin (*motion*) hesaplanması odometri tahmini olarak adlandırılır [5]. Bu bileşen için kullanılan iki temel yöntem; karşılaştırılacak veri çerçevelerindeki anahtar noktaların karşılıklı eşleştirilmesine dayalı nitelik tabanlı (*feature based*) yaklaşım ve çerçevelerdeki verilerin tamamının piksel bazında hata minimizasyonu yoluyla kullanımına dayalı yoğun (*dense*) yaklaşımdır [6]. Döngü kapama tespiti ise, robotun ortamda daha önce geçtiği bir bölgeden tekrar geçmekte olduğunun anlaşılmasıdır [7]. Robotun gerçekleştirdiği döngü kapamaların başarılı şekilde tespit edilmesi haritanın doğruluğu açısından son derece önemlidir. Robotun bulunduğu yeni bir bölgenin daha önce ziyaret edilmiş bir bölge olarak değerlendirilmesi, ya da zaten ulaşılmış bir bölgenin ortamda daha önce bulunulmamış bir alan şeklinde yorumlanması

sistemi başarısızlığa uğratabilecek düzeyde hatalardır. Döngü kapama tespiti için kullanılan genel yaklaşım, algılayıcıdan elde edilen çerçevelerle önceki gözlem verileri arasındaki görsel ilişkilerin mekân tanıma (*place recognition*) yöntemleriyle elde edilmesidir. Mekân tanıma probleminin kendine has zorluklarının dışında bir diğer problem ise, robotun ortamdaki gezinimi süresince artarak biriken gözlem verileri içerisinde yapılacak aramanın getirdiği maliyettir. Bu temel zorluklar sebebiyle, günümüzdeki gelişmiş RGB-D algılayıcı tabanlı SLAM yöntemlerinin büyük bir kısmı geniş ortamları haritalama yeteneğine sahip değildir.

Tez kapsamında, RGB-D algılayıcı tabanlı Eş Zamanlı Konum Belirleme ve Haritalama alanında çalışmalar gerçekleştirilmiştir. Yapılan çalışmaların değerlendirilmesi aşamasında, literatürdeki SLAM sistemlerinin başarımlarında kullanılan popüler veri kümeleri üzerinde elde edilen sonuçlar incelenmiştir. Hareket tahmini aşamasında veri çerçevelerinin nitelik tabanlı çıkarılması konusunda çeşitli analizler gerçekleştirilmiştir. Yapılan deneysel değerlendirmeler sonucunda doğruluk ve hız yönünden en etkili nitelik bulucu (*feature detector*) ve nitelik tanımlayıcı (*feature descriptor*) ikilisi tespit edilmiş ve ileriki çalışmalarda bu ikili kullanılmıştır. Döngü kapama tespiti için, genel ve yerel görüntü niteliklerinin birlikte kullanımına dayalı bir yöntem geliştirilmiştir. İki aşamalı şekilde çalışan yöntem, ilk adımda genel görüntü niteliklerine göre düşük maliyetle filtreleme yapmakta, sonrasında ise yerel niteliklere göre mekân tanıma gerçekleştirerek döngü kapama adaylarını belirlemektedir. Geliştirilen yöntem, küçük çaplı ortamlarda etkili sonuçlar üretebilen ancak geniş ortamlarda aynı performansı sergileyemeyen gelişmiş bir RGB-D SLAM sistemine entegre edilmiş ve sistem başarımlarını artırılarak yöntemin geniş ortamlarda da etkin şekilde çalışabilmesi sağlanmıştır. Sonraki çalışmada, geliştirilmiş olan döngü kapama tespiti yöntemi yeni bileşenlerle genişletilerek hem yöntemin hızı önemli derecede artırılmış hem de başarımları korunmuştur. Tez kapsamındaki son çalışmada ise, kapalı ortamları 3 boyutlu olarak haritalayabilen bir RGB-D SLAM sistemi geliştirilmiştir. *k*-SLAM adındaki sistem, küçük ortamlarda olduğu kadar geniş ortamlarda da etkili haritalama sonuçları üretebilmektedir. Sistem nitelik tabanlı hareket tahmini gerçekleştirmekte ve genel görüntü niteliklerinin sıradüzensel şekilde indekslenmesine dayalı bir döngü kapama tespiti mekanizması kullanmaktadır. Küçük, orta ölçekli ya da geniş ortamlarda benzer hızlarda

çalışabilen  $k$ -SLAM, bu yönüyle literatürdeki diğer yöntemlerden ayrılmaktadır. Özellikle geniş çaplı haritalama başarısında gelişmiş birçok SLAM sistemini geride bırakan sistem, CPU üzerinde gerçek zamanlı çalışabilmektedir.

Tezin devam eden kesimleri şu şekilde tasarlanmıştır:

İkinci bölümde SLAM problemi ele alınmıştır. SLAM kapsamındaki temel zorluklara ve kullanılan algılayıcılara dair detaylara yer verilmiştir. Problemin olasılıksal tanımı ve geliştirilen temel çözüm yöntemleri açıklanarak sahip oldukları avantaj ve dezavantajlar bazında değerlendirmeler yapılmıştır.

Üçüncü bölümde literatürde yer alan ilgili çalışmalara yer verilmiştir. Geliştirilmiş olan yöntemler temel olarak hareket tahmini yaklaşımlarına göre gruplandırılarak incelenmiştir. Bölümün son kısmında ise, bu alanda popüler olarak kullanılan veri kümeleri ve haritalama başarımının kıyaslanması için kullanılan hata metriğine dair ayrıntılar yer almaktadır.

Dördüncü bölümde, nitelik tabanlı haritalama sistemlerinde hareket tahmini aşamasında kullanılacak çeşitli nitelik bulucular ve nitelik tanımlayıcılar gelişmiş bir RGB-D SLAM sistemi üzerinde test edilmiştir. Gerçekleştirilen analizlerde sistem hızı ve harita doğruluğu birlikte değerlendirilerek, en etkili nitelik bulucu - nitelik tanımlayıcı kombinasyonu belirlenmiştir.

Beşinci bölümde, nitelik tabanlı SLAM sistemlerine kolayca entegre edilebilecek bir döngü kapama tespiti yöntemi geliştirilmiştir. Arama uzayını genel görüntü niteliklerini kullanarak daraltan ve yerel niteliklerin kullanımıyla döngü kapama adaylarını belirleyen yöntem ayrıntılı şekilde tanımlanmış ve entegre edildiği SLAM sistemi üzerinde değişik parametre kombinasyonlarına göre elde edilen deneysel sonuçlar yorumlanmıştır.

Altıncı bölümde, geliştirilmiş olan döngü kapama tespiti yöntemi sıradüzensel kümeleme ve uyarlanabilir eşikleme adındaki yeni bileşenler ile genişletilerek yöntemin daha hızlı ve verimli çalışması hedeflenmiştir. Bir önceki bölüme benzer şekilde, genişletilmiş yöntem aynı SLAM sistemine entegre edilerek bulunan deneysel sonuçlar analiz edilmiştir.

Yedinci bölümde, geliştirilmiş olan RGB-D algılayıcı kullanan 3B SLAM sistemi tanıtılmıştır. Sistemin çalışma mekanizması temel bileşenleriyle ayrıntılı olarak sunulmuştur. Popüler veri kümeleri üzerinde çok sayıda görüntü kaydı ile kapsamlı



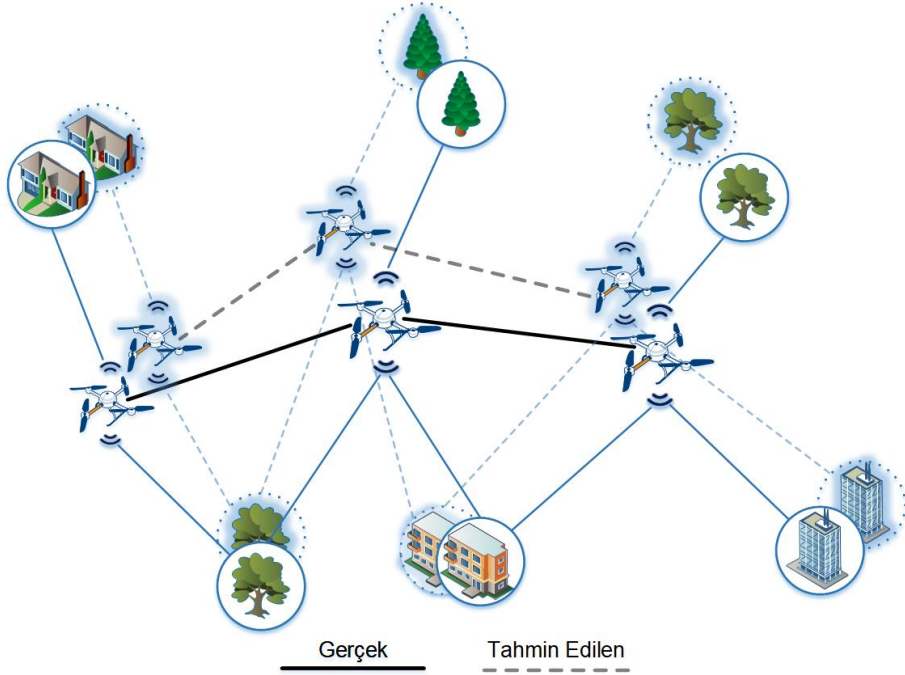
deneyler gerekleřtirilmiřtir. Elde edilen deneysel sonular kullanarak, literatürdeki diđer geliřmiř RGB-D tabanlı haritalama sistemleri ile karřılařtırma yapılmıřtır.

Sekizinci bölümde ise, tez alıřması kapsamında geliřtirilen yöntemlere dair elde edilen sonular irdelenmiřtir. Gelecekte ne gibi alıřmalar yapılabileceđi konusunda bilgi verilmiřtir.

## 2. EŞ ZAMANLI KONUM BELİRLEME VE HARİTALAMA

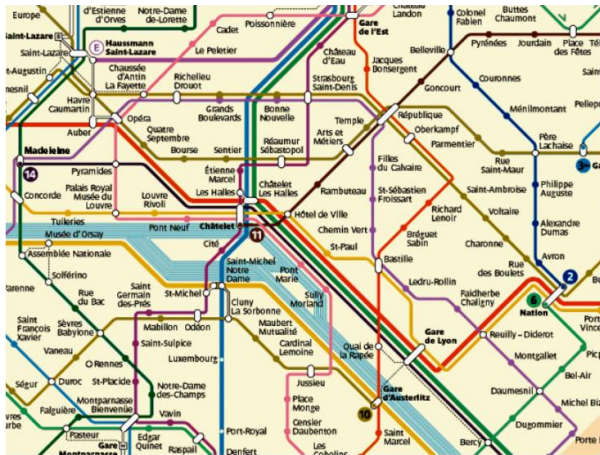
Eş Zamanlı Konum Belirleme ve Haritalama; bulunduğu ortamda hareket edebilen ve üzerindeki algılayıcılarla (kamera, lazer tarayıcı vb.) çevresini gözlemleyebilen bir robotun, aynı anda hem ortamın haritasını elde edebilmesi (*mapping*) hem de harita içerisinde konumunu belirleyebilmesi (*localization*) problemidir. SLAM problemi kapsamında robot, ortam hakkında herhangi bir ön bilgiye sahip olmadan, algılayıcılardan gelen verileri yorumlayarak ortamın haritasını çıkarmaktadır. Robot otonom hareket bağlamında, oluşturduğu haritayı kullanarak karar vermektedir [8].

Şekil 2.1'de SLAM problemini örnekleyen bir senaryo görülmektedir [9]. Şekilde yer alan insansız hava aracı (İHA) kendi konumunu kusursuz biçimde tespit edebilseydi, üzerindeki algılayıcının sağladığı gözlem verilerini yorumlayarak etraftaki yer göstericilerin (*landmark*; robotun gözlemleyebildiği herhangi bir varlık) pozisyonlarını bulabilecek ve haritayı belirli bir hata payı ile oluşturabilecekti. Diğer yandan, İHA ortamın hatasız bir haritasına sahip olabilseydi, bu haritayı kullanarak kendi konumunu belirli bir hata payı ile belirleyebilecekti. Ancak İHA kusursuz bir konum belirleme sistemine ya da hatasız bir haritaya sahip olmadığından, eş zamanlı olarak hem konumunu belirlemeli hem de haritayı oluşturmalıdır. Konum belirleme başarısı haritanın doğruluğuna, harita oluşturma başarısı ise konum verisinin doğruluğuna bağlıdır. Dolayısıyla hem tahmin edilen konum bilgisi hem de oluşturulan harita birbirleriyle ilişkili (*correlated*) olup belirli hata paylarına sahiptirler. Bu sebeple SLAM problemi olasılıksal yöntemlerle çözülmeye çalışılmaktadır.



Şekil 2.1. SLAM problemine dair örnek senaryo [9]

Oluşturulan harita, robotun ortamı modellemek için kullandığı herhangi bir gösterim olabilir. Bu gösterimler örneğin içerdiği veri türüne göre topolojik ya da geometrik haritalar şeklinde oluşturulabilir (Bkz. Şekil 2.2). Topolojik haritalar, kapsadıkları alanda yer alan bölgeler arasındaki ilişkileri göstermekte olup herhangi bir mesafe bilgisi içermezler. Geometrik haritalar ise, bölgeler arasındaki bağlantıların yanı sıra metrik uzaklık verilerini de içerirler. Diğer yandan ise haritalar nitelik tabanlı ya da hacimsel (*volumetric*) olabilir, 2 boyutlu ya da 3 boyutlu olarak oluşturulabilirler.



Topolojik Harita



Geometrik Harita

Şekil 2.2. Topolojik ve geometrik harita örnekleri [10]

## **2.1. SLAM Kapsamındaki Temel Zorluklar**

### **2.1.1. Algılayıcı Ölçüm Belirsizlikleri**

SLAM bağlamında karşılaşılan en temel problemlerden birisi, algılayıcılardan elde edilen verilerdeki gürültü ya da teknik kısıtlardan kaynaklı belirsizliklerdir. Algılayıcıların sağladığı ölçüm verilerindeki belirsizlikler, konum ve gözlem tahminlerine dair başarıyı düşürmekte, bu iki işlev arasındaki ilişkiden dolayı da hata payını gittikçe daha fazla arttırmaktadır.

### **2.1.2. Veri İlişkilendirme Problemi**

Bir diğer temel problem, veri ilişkilendirme (*data association*) problemi olarak adlandırılan, gözlemlenen yer göstericiler arasındaki ilişkilerin belirlenmesidir [11]. Gözlemlenmekte olan yer göstericinin gezinim sırasında zaten algılanmış olup olmadığına karar verebilme yeteneği, bir SLAM yöntemi için olmazsa olmaz gereksinimdir. Bu konuda gerçekleştirilecek ufak bir hata, konum tahmininde büyük bir kaymaya (*drift*) sebebiyet verebilir çünkü konum tahmini yer göstericilerin pozisyonlarına göre yapılmaktadır. Dolayısıyla konum tahmininde oluşabilecek büyük bir kayma, oluşturulan haritanın tamamen yanlış olmasına sebebiyet verebilir. Diğer taraftan veri ilişkilendirme yeteneği, döngü kapama tespiti açısından da önemli role sahiptir. Döngü kapama tespiti; robotun ortamda daha önce geçtiği bir bölgeden tekrar geçmekte olduğunun anlaşılması, yani algılanmakta olan mevcut sahnenin daha önce de algılanmış olduğunun belirlenebilmesi yeteneğidir. Herhangi bir yerden daha önce geçildiğinin tespit edilememesi durumunda, robot bu bölgeyi yeni bir bölge olarak değerlendirecek, bu durumda aynı bölge haritanın farklı kısımlarında birden fazla bulunacaktır. Tespitin başarılı yapılabilmesi durumunda ise o bölgedeki yer göstericilerin ve robotun konumları daha iyi kestirilebilmektedir. Veri ilişkilendirme yeteneği bir SLAM sistemi için en kritik bileşenlerden biri olduğu gibi, SLAM kapsamında çözülmesi en zor problemlerden de biridir.

### **2.1.3. Ortam Büyüklüğü**

Haritası oluşturulacak ortamın büyüklüğü SLAM başarımını etkileyen bir diğer faktördür. Haritalama sırasında gerek konum tahmini gerekse döngü kapama tespiti için, mevcut algılayıcı verilerinin yanında önceki veriler de kullanıldığından, ortam büyüdükçe işlenmesi gereken veri miktarı da gittikçe artmaktadır.

Dolayısıyla işlemsel yük bir noktadan sonra önemli derecede artış göstermektedir. Otonom robotlar için harita oluşturma işlemi gerçek zamanlı olması gereken bir işlemdir. Ayrıca robotun sahip olduğu sistem kaynakları (bellek gibi) kısıtlıdır. Dolayısıyla, kısıtlı kaynaklarla artan işlemsel yükün üstesinden gelinemesi gerçek zamanlı işlem için bir gerekliliktir. Ayrıca ortam genişledikçe kayma hataları daha da artmakta, döngü kapamaların tespit edilmesi daha da güçleşmekte, bundan dolayı harita oluşturma başarımları da gittikçe düşmektedir.

#### **2.1.4. Ortamın Dinamik Olması**

Ortamda robot dışında hareket eden herhangi bir nesne ya da nesnelerin olması, yani ortamın değişmesi haritalamayı oldukça zorlaştırmaktadır. Dinamik ortam problemi, hali hazırda SLAM bağlamında çözülmesi çok zor bir problemdir. Bu sebeple, literatürde yer alan SLAM algoritmalarının çok büyük bir kısmında, haritası oluşturulacak ortamın robot dışında statik olduğu kabul edilmektedir.

#### **2.2. SLAM Sistemlerinde Kullanılan Algılayıcılar**

SLAM sistemlerinin geliştirilmesinde, kullanılan algılayıcının türü büyük öneme sahiptir. Literatürde yer alan metotlarda, mesafe algılayıcıları (lazer ya da sonar) [12]–[14] ve kameraların (stereo ya da tek kamera) [15]–[17] birincil algılayıcı olarak kullanıldıkları görülmektedir.

Kameralar zengin veri kapasiteleri, kolay taşınabilir olmaları ve ucuzlukları sayesinde SLAM sistemlerinde yaygın olarak tercih edilmektedirler. Tek kamera kullanan yaklaşımlarda ölçek belirsizliği (*scale ambiguity*) önemli bir problemdir. Çünkü tek bir çerçeveden derinlik verisi elde edilmesi mümkün değildir, dolayısıyla haritanın mutlak ölçek değeri belirlenemediği için ekstra normalleştirme işlemleri gerçekleştirilmektedir [5]. Stereo kamera sistemlerinin kullanıldığı yöntemlerde ise, farklı açılardan kaydedilen görüntülerde ortak olan noktalar tespit edilerek renk verisinin yanında derinlik verisi de elde edilebilmektedir. Ancak derinlik verisinin güvenilir şekilde hesaplanabilmesi için ortamın dokulu (*textured*) olması gereklidir.

Diğer yandan, mesafe algılayıcısı kullanılarak geliştirilen yöntemlerde yalnızca derinlik bilgisi mevcut olduğundan, görsel verinin sağladığı avantajlarından yararlanamayan bu tarz yöntemler için veri ilişkilendirme probleminin çözümü oldukça zordur. Ayrıca mesafe algılayıcılarının maliyetleri yüksektir.

Microsoft Kinect ya da ASUS Xtion Pro gibi RGB-D algılayıcıların ortaya çıkışı ise SLAM uygulamalarının geliştirilmesi için önemli bir dönüm noktası olmuştur. RGB-D algılayıcıların en önemli avantajı renkli görüntü ile birlikte, gerçek zamanlı olarak derinlik haritasını da sunmalarıdır. Derinlik haritası, görüntüde yer alan nesnelerin uzaklık bilgilerini içeren bir matris olarak üretilmektedir. RGB-D kamera, ortama yaydığı kızılötesi ışınları, üzerindeki CMOS algılayıcı ile yakalayıp derinlik bilgilerini elde etmektedir. Bu sayede renk ve derinlik verileri birlikte kullanılarak hem veri ilişkilendirme problemi için daha etkili çözümler üretilebilmekte, hem de ortamdaki ışık seviyesinin değişimine bağlı problemlere karşı daha dayanıklı algoritmalar geliştirilebilmektedir.

### 2.3. SLAM Probleminin Olasılıksal Tanımı

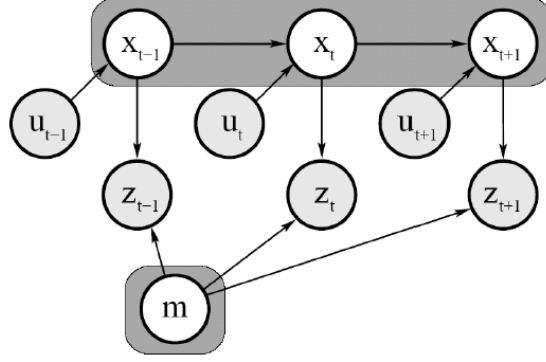
SLAM problemi olasılıksal biçimde aşağıdaki gibi formüle edilmektedir [1].

$$p(x_{0:T}, m \mid z_{1:T}, u_{1:T}) \quad (2.1)$$

Burada robotun  $t$  zamanındaki konumu  $x_t$  olarak gösterilmektedir. Dolayısıyla  $x_{0:T} = \{x_0, x_1, x_2, \dots, x_T\}$  robotun ortamdaki gezinimi süresince bulunduğu konumlar dizisi, yani yapmış olduğu yoldur.  $m$  ortam haritasını,  $z_{1:T} = \{z_1, z_2, z_3 \dots, z_T\}$  robotun yapmış olduğu gözlemleri (örneğin kamera görüntüleri) ifade eder.  $u_{1:T} = \{u_1, u_2, u_3 \dots, u_T\}$  ise gezinim boyunca robotun gerçekleştirdiği hareketler dizisini yani odometri verilerini simgeler.

Robotun elde ettiği gözlem ve odometri verileri hatadan bağımsız değildir, çünkü algılayıcı verileri gürültü içermektedir. Dolayısıyla, belirsizlik içeren bu problem olasılıksal yaklaşımla ele alınmaktadır. Eşitlik 2.1'deki dağılımla ifade edildiği üzere, gözlem ve odometri verileri kullanılarak, robotun bulunduğu konumlar ve haritanın sonsal olasılığı (*posterior probability*) tahmin edilmektedir. Bu işleme aynı zamanda *tam (full) SLAM* de denilmektedir.

Robotun katettiği yolun tamamı yerine sadece güncel konumunun ve haritanın o anki halinin tahmin edilmesi işlemi ise *çevrimiçi (online) SLAM* olarak adlandırılmaktadır. Çevrimiçi SLAM'de robotun daha önce bulunduğu konumlar göz ardı edilerek, sadece mevcut konumu tahmin edilmeye çalışılmaktadır. Tez çalışmaları kapsamında, geniş ortamların haritalanması için daha uygun olduğundan, tam SLAM üzerinde çalışılmıştır. Dolayısıyla robotun ortamdaki hareketi boyunca aldığı tüm yol ele alınmıştır.



Şekil 2.3. SLAM probleminin grafiksel gösterimi [3]

Şekil 2.3'teki grafiksel model, SLAM problemindeki değişkenler arası bağımlılıkları göstermektedir. Örneğin  $x_{t+1}$  konumu,  $x_t$  konumuna ve  $u_{t+1}$  odometri komutuna bağlıdır. Çünkü robot, bulunduğu konumdan ( $x_t$ ) bir sonraki konuma ( $x_{t+1}$ ) belirli bir hareket ( $u_{t+1}$ ) yaparak geçmektedir. Diğer yandan  $z_t$  gözlemi ise robotun bulunduğu  $x_t$  konumuna ve haritaya ( $m$ ) bağlıdır. Yani robotun herhangi bir nesneyi nasıl gözlemlediği, yer göstericilerin ortamda nerelerde bulunduğu (harita) ve robotun haritanın neresinde olduğuna (konum) göre şekillenmektedir.

SLAM problemi için geliştirilen yöntemlerde iki temel model kullanılmaktadır; hareket modeli (*motion model*) ve gözlem modeli (*observation model*).

- Hareket modeli robotun nasıl hareket ettiğini tanımlar.

$$p(x_t | x_{t-1}, u_t) \quad (2.2)$$

Robotun şu anki konumuna ( $x_t$ ) ait dağılımın, bir önceki konum ( $x_{t-1}$ ) ve mevcut odometri komutu ( $u_t$ ) kullanılarak nasıl hesaplanacağı hareket modeli ile belirlenir. Robot bir odometri komutuyla hareket ederek yeni bir konuma geçtiğinde, odometri ölçümlerindeki belirsizliklerden dolayı yeni konumu yüzde yüz doğrulukla kestirilememektedir. Dolayısıyla robotun eski konumundan bulunduğu yeni konuma geçişini ifade eden olasılıksal dağılım, hareket modeli kullanılarak tanımlanır.

- Gözlem modeli gözlemin nasıl yorumlanacağını ifade eder.

$$p(z_t | x_t, m_t) \quad (2.3)$$

Robotun konumu ( $x_t$ ) ve ortam modeli ( $m_t$ ) kullanılarak, mevcut gözlemin ( $z_t$ ) olasılıksal dağılımının hesaplanması gözlem modeli ile gerçekleştirilir. Algılayıcı ölçümlerindeki belirsizlikler bu model ile tanımlanarak, konum ve

harita verilerine göre gözlemin olasılığı belirlenir. Gözlem modeli, kullanılan algılayıcının türüne göre değişmektedir.

Bu iki model için uygun gösterimler üretilerek SLAM problemi çözülmeye çalışılmaktadır [9]. Örneğin hareket ve gözlem modelleri için doğrusal (*linear*) / doğrusal olmayan (*non-linear*) gösterimlerin kullanılması, dağılım fonksiyonunun Gauss / Gauss olmayan şekilde seçilmesi, parametrik / parametrik olmayan biçimlerin kullanılması gibi tercihler geliştirilen SLAM yaklaşımını şekillendirmektedir.

## 2.4. SLAM Problemi İçin Geliştirilen Temel Yöntemler

SLAM probleminin çözümü için geliştirilen yöntemler 3 ana başlıkta ele alınmaktadır. Bu temel yöntemler; Kalman Filtresi (*Kalman Filter*), Parçacık Filtresi (*Particle Filter*) ve Çizge Tabanlı (*Graph Based*) yöntemlerdir.

### 2.4.1. Kalman Filtresi Yöntemi

Bu yöntemde SLAM problemi, Kalman Filtresi kullanılarak durum tahmini (*state estimation*) yapılması şeklinde ele alınır [10]. Hareket modeli ve gözlem modelinin doğrusal fonksiyonlar olduğu ve Gauss dağılımına sahip oldukları varsayılır [3]. Dolayısıyla hareket ve gözlem modelleri aşağıdaki gibi tanımlanır [9];

$$p(x_t | x_{t-1}, u_t) \Leftrightarrow x_t = f(x_{t-1}, u_t) + \epsilon_t \quad (2.4)$$

$$p(z_t | x_t, m_t) \Leftrightarrow z_t = g(x_t, m_t) + \delta_t \quad (2.5)$$

Burada  $f(\cdot)$  ve  $g(\cdot)$  fonksiyonları doğrusal olmak üzere,  $\epsilon_t$  ve  $\delta_t$  sırasıyla hareket ve ölçüm hatalarını ifade eden,  $R_t$  ve  $Q_t$  kovaryansları (*covariance*) ile normal dağılıma sahip rastgele değişkenlerdir.

Kalman Filtresi kullanılarak problemin ele alınması için hareket ve gözlemin doğrusal olması gerekir, ancak dinamik sistemlere ait gerçek senaryolar için genellikle doğrusal olmayan fonksiyonlar geçerlidir. Doğrusal olmayan hareket ve gözlem modelleri ise Gauss olmayan dağılımlara sebep olur, bu da Kalman Filtresi uygulanmasının önünde engel teşkil eder. Dolayısıyla problem ele alınırken yerel doğrusallaştırma (*local linearization*) ile  $f(\cdot)$  ve  $g(\cdot)$  fonksiyonları doğrusal hale getirildikten sonra Kalman Filtresi uygulanır. Bu yöntem, Genişletilmiş Kalman Filtresi (*Extended Kalman Filter*) yöntemi olarak adlandırılır.



Yöntemde harita nitelik tabanlı oluşturulmakta, yani yer göstericilerin konumlarını içermektedir. Robotun ve yer göstericilerin tahmin edilen konumlarına ait veriler bir durum vektöründe (*state vector*) tutulmaktadır. Yapılan tahminlerdeki belirsizliklerin ifade edilmesi için ise, robotun ve yer göstericilerin durum tahminleri arasındaki ilişkileri içeren kovaryans matrisi tanımlanmıştır [1]. Bu yöntem genellikle çevrimiçi SLAM için kullanılmakta ve aşağıdaki gibi ifade edilmektedir;

$$p(x_t, m | z_{1:T}, u_{1:T}) \sim \mathcal{N}(\mu_t, \Sigma_t) \quad (2.6)$$

Burada  $\mu_t$  durum vektörünü,  $\Sigma_t$  ise kovaryans matrisini ifade eder. Bu veri yapıları, robotun gezinimi süresince yaptığı gözlemlere göre Genişletilmiş Kalman Filtresi ile güncellenmektedir. Haritalamanın 2 boyutlu uzayda yapıldığı varsayımıyla  $\mu$  ve  $\Sigma$ , Şekil 2.4'teki gibi oluşmaktadır. Burada  $x, y, \theta$  robotun konumunu (koordinatlar:  $x$  ve  $y$ , açısız yönelim:  $\theta$ ),  $m_{n,x}$  ve  $m_{n,y}$  ise  $n$ 'inci yer göstericinin koordinatlarını ifade etmektedir. Kovaryans matrisinde ise örneğin  $\sigma_{m_{1,x},x}$ , yer gösterici  $m_1$ 'in konumunun  $x$  koordinatı ile robotun konumunun  $x$  koordinatı arasındaki ilişkiyi içerir.

$$\underbrace{\begin{pmatrix} x \\ y \\ \theta \\ m_{1,x} \\ m_{1,y} \\ \vdots \\ m_{n,x} \\ m_{n,y} \end{pmatrix}}_{\mu} \quad \underbrace{\begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{x\theta} & \sigma_{xm_{1,x}} & \sigma_{xm_{1,y}} & \dots & \sigma_{xm_{n,x}} & \sigma_{xm_{n,y}} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{y\theta} & \sigma_{ym_{1,x}} & \sigma_{ym_{1,y}} & \dots & \sigma_{ym_{n,x}} & \sigma_{ym_{n,y}} \\ \sigma_{\theta x} & \sigma_{\theta y} & \sigma_{\theta\theta} & \sigma_{\theta m_{1,x}} & \sigma_{\theta m_{1,y}} & \dots & \sigma_{\theta m_{n,x}} & \sigma_{\theta m_{n,y}} \\ \sigma_{m_{1,x}x} & \sigma_{m_{1,x}y} & \sigma_{\theta} & \sigma_{m_{1,x}m_{1,x}} & \sigma_{m_{1,x}m_{1,y}} & \dots & \sigma_{m_{1,x}m_{n,x}} & \sigma_{m_{1,x}m_{n,y}} \\ \sigma_{m_{1,y}x} & \sigma_{m_{1,y}y} & \sigma_{\theta} & \sigma_{m_{1,y}m_{1,x}} & \sigma_{m_{1,y}m_{1,y}} & \dots & \sigma_{m_{1,y}m_{n,x}} & \sigma_{m_{1,y}m_{n,y}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \sigma_{m_{n,x}x} & \sigma_{m_{n,x}y} & \sigma_{\theta} & \sigma_{m_{n,x}m_{1,x}} & \sigma_{m_{n,x}m_{1,y}} & \dots & \sigma_{m_{n,x}m_{n,x}} & \sigma_{m_{n,x}m_{n,y}} \\ \sigma_{m_{n,y}x} & \sigma_{m_{n,y}y} & \sigma_{\theta} & \sigma_{m_{n,y}m_{1,x}} & \sigma_{m_{n,y}m_{1,y}} & \dots & \sigma_{m_{n,y}m_{n,x}} & \sigma_{m_{n,y}m_{n,y}} \end{pmatrix}}_{\Sigma}$$

Şekil 2.4. Genişletilmiş Kalman Filtresi yönteminde kullanılan durum vektörü ( $\mu$ ) ve kovaryans matrisi ( $\Sigma$ ) [10].

Yöntem temel olarak aşağıdaki adımlardan oluşmaktadır [10];

1. Robotun yeni konumunun, uygulanan odometri komutu kullanılarak tahmin edilmesi.
2. Robot konumuna göre, yapılması beklenen gözlemin tahmin edilmesi.
3. Gerçek gözlemin elde edilmesi.
4. Gözlemlenmekte olan yer göstericilerin, daha önce gözlemlenmiş olan yer göstericilerle ilişkisinin tespit edilmesi (veri ilişkilendirme).

5. Tahmin edilen gözlem ile gerçekte elde edilen gözlem arasındaki farkın hesaplanması.

6. Durum vektörünün ve kovaryans matrisinin güncellenmesi.

Bu şekilde, robot ortamda gezindikçe yeni yer göstereciler gözlemlenmekte ve bunlara dair konum verileri durum vektörüne eklenmektedir. Kovaryans matrisi de aynı şekilde, yeni yer göstereciler ile robot ve mevcut yer göstereciler arasındaki ilişki verileri ile genişlemektedir. Dolayısıyla  $N$  yer gösterici sayısı olmak üzere durum vektörü  $3 + 2N$  boyutunda iken, kovaryans matrisi ise  $(3 + 2N) \times (3 + 2N)$  boyutundadır. Dolayısıyla işlem maliyeti, yer gösterecilerin sayısına göre karesel (*quadratic*) şekilde artmaktadır ( $O(N^2)$ ).

Genişletilmiş Kalman Filtresi yöntemi geniş ortamlar ve gerçek zamanlı haritalama için uygun bir yöntem değildir. Gezinim boyunca gözlemlenen yer gösterecilerin sayısının sürekli artıyor olması işlem yükünü önemli oranda arttırmakta, dolayısıyla uygulanabilirliği azaltmaktadır.

Bu yaklaşıma benzer olarak; Kokusuz Kalman Filtresi (*Unscented Kalman Filter*), Genişletilmiş Bilgisel Filtre (*Extended Information Filter*) ve Seyrek Genişletilmiş Bilgisel Filtre (*Sparse Extended Information Filter*) şeklinde Kalman türevi yöntemler de literatürde mevcuttur. Kokusuz Kalman Filtresi yöntemi, doğrusallaştırma işleminin daha başarılı yapılabilmesi için geliştirilmiştir. Bu yöntem doğrusal olmayan modeller için Genişletilmiş Kalman Filtresi'nden daha başarılı sonuç üretebilmektedir ancak hızı daha düşüktür. Genişletilmiş Bilgisel Filtre kullanımında ise Gauss dağılımı farklı bir uzayda ele alınır. Kovaryans matrisi yerine bunun tersi olan bilgi matrisi (*information matrix*), durum vektörü yerine ise bilgi matrisi ile çarpımı olan bilgi vektörü (*information vector*) kullanılır. Genişletilmiş Kalman Filtresi'nin benzer şekilde bilgi uzayına adapte edilerek uygulanması, Genişletilmiş Bilgisel Filtre olarak adlandırılır. Tahmin aşamasında Kalman'dan daha yavaş, düzeltme aşamasında ise daha hızlıdır. Ancak bilgi uzayına geçiş işlemleri maliyetlidir ve Kalman ile benzer başarı sergilemektedir. Seyrek Genişletilmiş Bilgisel Filtre yönteminde ise, bilgi matrisindeki zayıf ilişkiler göz ardı edilerek matris seyrek şekilde tutulup işlemlerin hızlandırılması hedeflenmektedir. Bu yöntem, Genişletilmiş Kalman Filtresi'ne göre daha hızlı çalışmaktadır ancak sistem başarımı daha düşüktür.

Sonuç olarak, gerek başarım gerekse çalışma hızı açısından günümüzde SLAM uygulamaları için Kalman Filtresi türevi yöntemler pek tercih edilmemektedir.

#### 2.4.2. Parçacık Filtresi Yöntemi

Bir diğer temel yaklaşımda ise Parçacık Filtresi kullanımıyla sistem durumu tahmin edilir [1]. Bu yöntemde dağılım parametrik olmayan gösterimle ele alınmaktadır. Dolayısıyla Kalman Filtresi tabanlı yöntemlerdeki gibi modellerin Gauss dağılımına sahip olması zorunluluğu yoktur. Filtrede yer alan her bir parçacık ile sistem durumuna dair bir tahmin yapılır. Sistemin sonsal olasılığı, parçacıkların yaptıkları tahminlere göre ağırlıklandırılması (*weighting*) ile elde edilir. Sistemdeki parçacıklar kümesi, içerdiği her bir parçacık ile aşağıdaki gibi ifade edilir [10];

$$S = \{\{x^{[j]}, w^{[j]}\}\}_{j=1, \dots, J} \quad (2.7)$$

Burada  $j$  parçacığı  $x^{[j]}$  ile, parçacığın ağırlığı ise  $w^{[j]}$  ile ifade edilmektedir. Parçacık Filtresi uygulanırken bir teklif dağılımı (*proposal distribution*) belirlenir, parçacıklar bu dağılımdan örneklenir (*sampling*) ve hedef dağılıma (*target distribution*) ne kadar yakın sonuç ürettiğine bağlı olarak parçacıklar ağırlıklandırılır. Sonrasında ise yeniden örnekleme (*resampling*) ile yüksek ağırlıklı parçacıklardan seçim yapılarak sonraki yinelemede (*iteration*) hedef dağılıma daha yakın sonuç elde edilmesi amaçlanır.

Parçacık Filtresi'nin SLAM kapsamında konum belirleme için uygulanan popüler bir örneği Monte Carlo Localization (MCL) olarak adlandırılır. Bu yaklaşımda her bir parçacık, robot konumuna dair bir tahmin gerçekleştirir. Teklif dağılımı, hareket modeli olarak seçilir ve gözlem modeli kullanılarak ağırlıklandırma yapılır. Dolayısıyla bu yaklaşım aşağıdaki gibi ifade edilebilir [10];

$$x_t^{[j]} \sim p(x_t | x_{t-1}^{[j]}, u_t) \quad (2.8)$$

$$w_t^{[j]} = \frac{\text{hedef\_dağılım}(x_t^{[j]})}{\text{teklif\_dağılım}(x_t^{[j]})} \propto p(z_t | x_t, m_t) \quad (2.9)$$

Odometri komutu işletilerek hareket modeli uygulanır ve modelden seçilen parçacıkların ürettiği konum sonuçları ( $x_t^{[j]}$ :  $j$  parçacığı için  $t$  zamanındaki robot konumu) elde edilir. Sonrasında ise gözlem modeli kullanılarak hedef dağılıma ne kadar yaklaşıldığına göre parçacıkların ağırlık değerleri belirlenir. Yeniden örnekleme aşamasında, daha yüksek ağırlığa sahip (daha başarılı konum tahmini

yapmış olan) parçacıklar seçilerek bir sonraki robot hareketinde bu parçacıkların tahminleri kullanılır.

Parçacık Filtresi düşük boyutlu uzaylarda etkili sonuçlar üretebilmektedir. Diğer yandan boyut sayısı arttıkça parçacık sayısının da artırılması gerekmektedir. Haritalama işleminde ise hem robotun hem de çok sayıda yer göstericinin konumlarının ele alınması gerektiğinden çok yüksek boyutlu uzay söz konusudur. Örneğin nitelik tabanlı bir haritada yüz binlerce yer gösterici bulunabilir. Dolayısıyla SLAM bağlamında haritalama işlemi için Parçacık Filtresi'nin direkt olarak uygulanması pek mümkün değildir [3]. Bu sebeple, parçacıklar kullanılarak sadece robotun konumları tahmin edilmekte ve her bir parçacığın ürettiği sonuca (robotun aldığı yol) göre farklı bir harita oluşturulmaktadır. İşlem sonucunda, gerçeğe en yakın yol tahminini gerçekleştiren parçacığın ürettiği verilere göre harita elde edilmektedir.

Bu yaklaşım Rao-Blackwellization tekniği ile aşağıdaki gibi uygulanmaktadır [10];

$$p(x_{0:T}, m_{1:M} | z_{1:T}, u_{1:T}) = p(x_{0:T} | z_{1:T}, u_{1:T}) p(m_{1:M} | x_{0:T}, z_{1:T}) \quad (2.10)$$

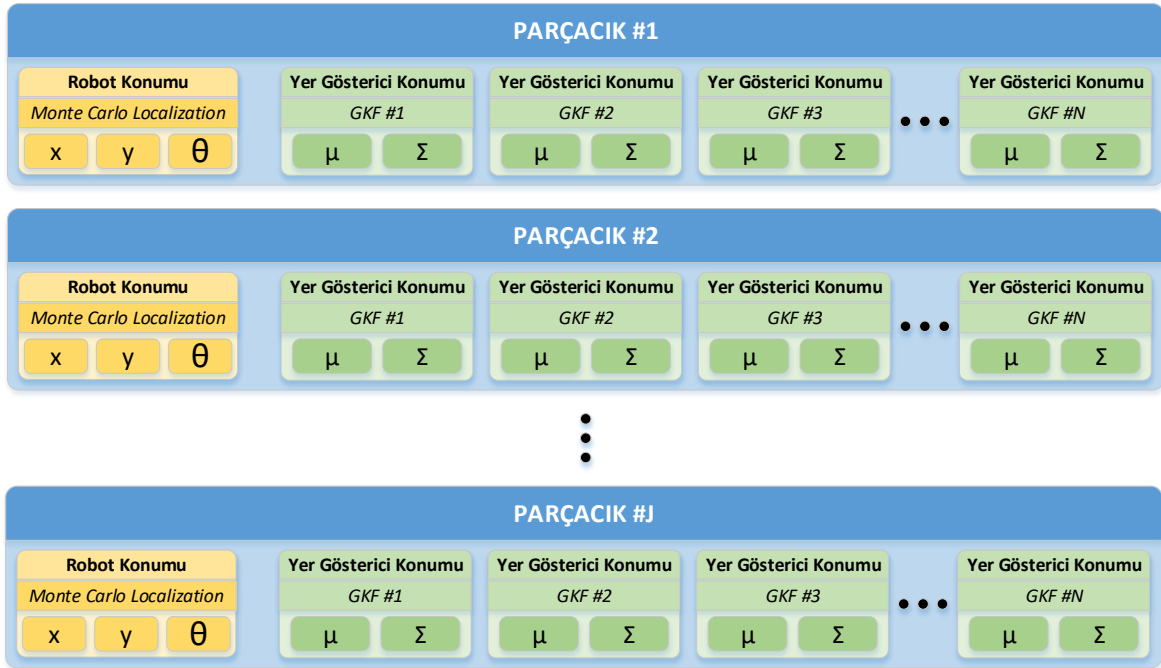
Buradaki amaç, yüksek maliyet gerektiren bir olasılıksal işlemin birbirlerinden bağımsız daha küçük parçalar halinde ele alınarak daha hızlı şekilde gerçekleştirilmesidir. Yer göstericilerin robot konumuna göre birbirlerinden bağımsız olmaları Rao-Blackwellization tekniğinin uygulanmasına imkân vermektedir. Bu teknikle dağılım çarpanlarına ayrılarak, ilk çarpan yani robotun aldığı yol ( $x_{0:T}$ ) Parçacık Filtresi uygulanarak tahmin edilmekte, ikinci çarpan yani yer gösterici konumları ( $m_{1:M}$ ) ise her bir parçacıkta hızlı şekilde hesaplanmaya çalışılmaktadır.

Bu yaklaşımın temel uygulaması FastSLAM algoritmasıdır [18]. Bu algorithmada her bir parçacık içerisinde robot konumu MCL yaklaşımı ile belirlenir. Bunun yanı sıra, parçacıklar içerisinde her bir yer gösterici için  $2 \times 2$ 'lik Genişletilmiş Kalman Filtresi tutulmaktadır. Bu şekilde her bir yer göstericinin konumu ayrı bir Kalman Filtresi ile tahmin edilmektedir. Şekil 2.5'te de görüldüğü üzere 2 boyutlu uzayda haritalama için, robot konumunu ifade eden  $x$  ve  $y$  koordinatları ile  $\theta$  açısall yönelimi MCL yaklaşımı ile tahmin edilmektedir. Her bir yer gösterici için ise, yine 2 boyutlu düzlemdeki konumunu gösteren  $x$  ve  $y$  koordinatları durum vektörü  $\mu$ 'de, koordinatlar arasındaki ilişkiler ise  $2 \times 2$ 'lik kovaryans matrisi  $\Sigma$ 'da tutulmaktadır.

Dolayısıyla yapılan tahminlerin gerçeğe yakınlığına göre parçacıkların ağırlıklandırılması aşağıdaki gibi ifade edilebilir [1];

$$w_t^{[j]} := \mathcal{N}(z_t | x_t^{[j]}, \mu_{t,n}^{[j]}, \Sigma_{t,n}^{[j]}) \quad (2.11)$$

Burada  $j$  parçacığına  $t$  zamanında verilen ağırlık  $w_t^{[j]}$ , parçacığın  $n$ 'inci yer gösterici'nin konumu için yaptığı tahmine dair durum vektörü  $\mu_{t,n}^{[j]}$ , kovaryans matrisi ise  $\Sigma_{t,n}^{[j]}$  ile gösterilmektedir.



Şekil 2.5. FastSLAM algoritmasının yapısı (GKF: Genişletilmiş Kalman Filtresi)

Yöntemin temel adımları aşağıdaki gibi özetlenebilir;

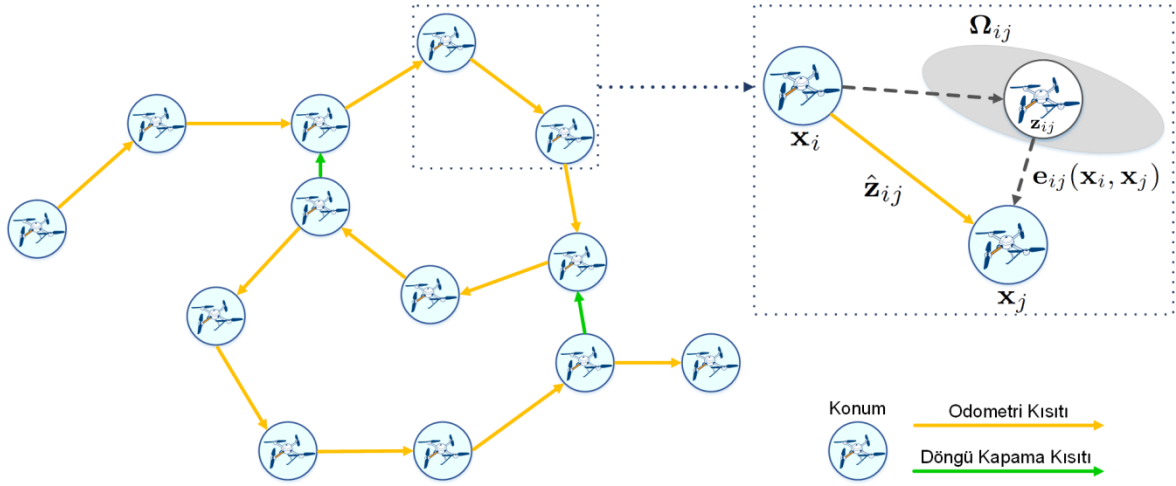
1. Her bir parçacık için;
  - 1.1. Odometri komutunun yerine getirilmesi ve uygulanan hareket modeline göre konum tahmininin elde edilmesi.
  - 1.2. Gözlemlenen yer göstericilerin konumlarının tahmin edilmesi.
  - 1.3. Parçacığın tahmin ettiği gözlemlerle gerçekte elde edilen gözlemlerin ne kadar örtüşüğünün hesaplanması ve parçacığın ağırlıklandırılması.
  - 1.4. Daha önce de gözlemlenmiş olan yer göstericilere ait Kalman Filtre'lerinin güncellenmesi.
  - 1.5. Yeniden örnekleme ile daha yüksek ağırlıklı parçacıkların seçilmesi.

Yöntemin Kalman Filtresi tabanlı SLAM yöntemlerine göre önemli bir avantajı, veri ilişkilendirmenin parçacık bazında yapılabilmesidir. Her bir parçacık yer göstericiler arasındaki ilişkileri kendisi belirlediğinden, daha başarılı veri ilişkilendirme daha başarılı konum tahmini anlamına gelmektedir. Dolayısıyla daha iyi veri ilişkilendirme yapan parçacıklar dolaylı olarak daha yüksek ağırlık elde etmektedir. Bir diğer avantajı ise daha az işlem yükü gerektirmesidir. Yer göstericilerin ikili arama ağacı (*binary search tree*) gibi bir veri yapısında tutulması durumunda,  $J$  parçacık sayısı ve  $N$  yer gösterici sayısı olmak üzere maliyet  $O(J \log N)$ 'dir. Ancak ortam genişledikçe, daha başarılı tahmin yapılabilmesi için daha fazla parçacık gerekmektedir. Bu kapsamda, FastSLAM 2.0 [19] teklif dağılımını belirlerken hareket modelinin yanı sıra algılayıcı gözlemini de kullanır. Bu sayede daha başarılı konum tahmini yapılabildiği için daha az sayıda parçacık kullanılarak daha iyi sonuç elde edilebilmektedir.

### 2.4.3. Çizge Tabanlı Yöntem

Bu yöntemde SLAM problemi, bir hata fonksiyonunun en küçük kareler (*least squares*) yöntemi ile optimizasyonu olarak tanımlanmaktadır [20]. Diğer yöntemlerde robot konumuyla birlikte ortamdaki yer göstericilerin pozisyonları da tahmin edilmeye çalışılırken, çizge tabanlı yaklaşımda yalnızca robotun bulunduğu konumlar üzerinden gittiği yol tahmin edilerek harita oluşturulabilmektedir. Bu sayede yer gösterici konum tahminlerinin getirdiği işlemsel yükten kurtularak daha hızlı ve verimli algoritmalar geliştirilebilmektedir.

Çizgenin düğümlerinde robotun ortamda hareket ederken bulunduğu konumlar ve o konumlarda yapılmış olan gözlemler tutulmakta, düğümler arasında yer alan kenarlar ise birbirlerine bağladıkları konumlar arasındaki uzamsal kısıtları içermektedirler (Bkz. Şekil 2.6). Sahip olduğu özelliklerle ilintili olarak bu çizge yapısına konum çizgesi (*pose graph*) adı verilmektedir [21]. Yöntemde, çizge üzerinde gerçekleştirilen optimizasyon vasıtasıyla hata minimize edilerek, düğümlerde yer alan robot konumları için en yüksek olasılıklı değerler belirlenmeye çalışılmaktadır. Dolayısıyla yöntemin temel amacı, robotun ortamda yapmış olduğu yol için bir maksimum olabilirlik çözümü (*maximum likelihood solution*) bulunmasıdır.



Şekil 2.6. Konum çizgesinin yapısı

Şekil 2.6'da görüldüğü üzere konum çizgesi, robotun ortamdaki hareketi neticesinde eklenen yeni düğümler ve kenarlar ile genişlemektedir. Yöntem temel olarak şu şekilde ifade edilebilir [21];

$x = (x_1, x_2, \dots, x_r)^T$  robotun ortamda bulunduğu konumlardan oluşan bir vektör olsun.  $z_{ij}$  ve  $\Omega_{ij}$  ise sırasıyla,  $x_i$  ve  $x_j$  konumlarına dair gözlemlerin birbiriyle maksimum şekilde örtüşmesini sağlayan dönüşüme (*transformation*) ait ortalama (*mean*) ve bilgi matrisi olsun.  $\hat{z}_{ij}$  ise çizgenin güncel durumuna göre  $x_i$  ve  $x_j$  konumları arasındaki tahmin edilen ölçümü ifade etsin. Yani daha açık ifadeyle  $z_{ij}$ ,  $x_i$  konumundan  $x_j$ 'nin nasıl gözlemlendiğini  $\Omega_{ij}$  belirsizliğiyle ifade ederken,  $\hat{z}_{ij}$  ise ilgili gözlemin çizgeye göre tahmin edilen değerini içersin. Gerçekte yapılan gözlemler tahmin edilen arasındaki fark ise  $e_{ij}(x_i, x_j)$  hata fonksiyonu ile tanımlansın;

$$e_{ij}(x_i, x_j) = z_{ij} - \hat{z}_{ij} \quad (2.12)$$

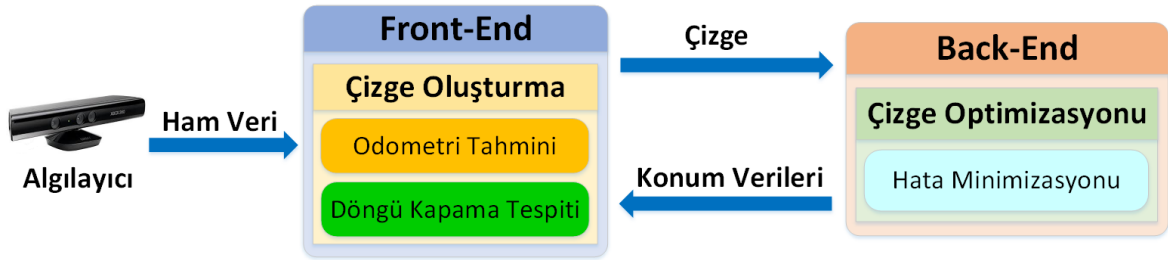
$C$  kümesi, aralarında herhangi bir uzamsal kısıt  $z$  bulunan (birbirine bağlı olan) düğümlere ait indis çiftlerini içersin. Bütün kısıtlara dair tüm negatif log olabilirlik (*negative log likelihood*) aşağıdaki gibidir;

$$F(x) = \sum_{(i,j) \in C} e_{ij}^T \Omega_{ij} e_{ij} \quad (2.13)$$

Çizge tabanlı yöntemin temel mantığı, negatif log olabilirlik  $F(x)$ 'i minimize eden optimal robot konumu konsigürasyonunun bulunmasıdır. Yani robotun yapmış olduğu yola dair en ideal sonuç ( $x^*$ ), maksimum olabilirlik yaklaşımı içerisinde hata minimize edilerek elde edilmeye çalışılmaktadır;

$$x^* = \operatorname{argmin}_x F(x) \quad (2.14)$$

Çizge tabanlı bir SLAM sistemi iki temel bileşenden oluşur; *front-end* ve *back-end* [22]. Çizgenin oluşturulması görevini front-end üstlenirken, back-end ise çizgenin optimize edilmesinden sorumludur (Bkz. Şekil 2.7). Front-end, algılayıcıdan gelen ham veriyi işleyerek çizgeyi yeni düğüm ve kenarlar ile genişletir. Back-end ise çizge optimizasyonunu, Eşitlik 2.14'ü çözerek yani Eşitlik 2.13'teki doğrusal olmayan hata fonksiyonunu minimize ederek gerçekleştirir.



Şekil 2.7. Çizge tabanlı SLAM yaklaşımı

Front-end, çizgeyi iki temel işlemlerle oluşturur; odometri tahmini ve döngü kapama tespiti. Odometri tahmini, algılayıcıdan art arda gelen veri çerçeveleri karşılaştırılarak aralarındaki ikili (*pairwise*) dönüşümün hesaplanması yoluyla gerçekleştirilir. Algılayıcıdan gelen her bir yeni çerçeve için odometri tahmini yapılarak robotun zaman içindeki hareketi hesaplanır. Böylece, robotun hareketi esnasında arda ardına bulunduğu konumlara ait çizgedeki düğümler arasına, hesaplanan odometri kısıtlarını içeren kenarlar eklenir (Bkz. Şekil 2.6'daki odometri kısıtı bulunan kenarlar).

Algılayıcının sağladığı verilerdeki belirsizlikler, hareket bulanıklığı (*motion blur*), ortamdaki aydınlatma koşulları gibi faktörler odometri tahmininin başarımını negatif yönde etkilemektedirler. Bu sebeple, tahmin edilen robot konumları zamanla kaymakta, dolayısıyla hesaplanan robot yolundaki kayma (*trajectory drift*) haritanın hatalı şekilde oluşturulmasına sebep olmaktadır.

Döngü kapamaların tespit edilmesi hesaplanan robot yolundaki kayma hatasının azaltılmasını sağlamaktadır. Robotun ortamda daha önce bulunduğu bir bölgeden tekrar geçmesi halinde; çizgede bu iki konumu ifade eden ilgili düğümlere ait veri çerçeveleri arasındaki ikili dönüşüm hesaplanarak, düğümler arasına döngü kapama kısıtını içeren kenar eklenir (Bkz. Şekil 2.6'daki döngü kapama kısıtı bulunan kenarlar). Back-end döngü kapama kısıtlarından yararlanarak çizgeyi



optimize eder. Dolayısıyla döngü kapamaların doğru şekilde tespit edilmesi, tahmin edilen robot yolundaki hatanın back-end tarafından optimizasyonla azaltılabilmesi için kritik öneme sahiptir. Back-end bu şekilde haritanın genel tutarlılığını (*global consistency*) sağlamaktadır.

Bu şekilde, algılayıcıdan gelen veri çerçeveleri front-end tarafından yorumlanarak çizgeye yeni düğümler ve kenarlar eklenmekte, yani robotun hareketiyle çizge büyümektedir. Çizgedeki hata ise back-end tarafından minimize edilerek robotun ortamda yapmış olduğu yola dair gerçeğe en yakın konumlar dizisi elde edilmeye çalışılmaktadır. Konumlar belirlendikten sonra; robotun bulunduğu her bir konumdayken gerçekleştirdiği gözlem (çizgedeki o konuma ait düğümde mevcut olduğundan), konuma göre ilgili dönüşüm uygulanarak ortak bir koordinat sisteminde birleştirilmekte ve ortam haritası elde edilmektedir.

Çizge tabanlı yöntem doğruluk, hız ve verimlilik açısından önemli avantajlara sahip olması sebebiyle SLAM probleminin çözümü için popüler şekilde kullanılmaktadır.  $N$  çizgedeki kenar sayısı olmak üzere, yöntemin işlem maliyeti kenar sayısı ile doğru orantılı yani  $O(N)$ 'dir. Yöntemde, yer gösterici konumları ele alınmadan sadece robot konumlarının hesaplanabiliyor olması işlemsel yükü önemli oranda azaltmaktadır. Dolayısıyla yoğun algılayıcı verisi işlenerek geniş ortamların haritalanabilmesi bu yöntemle daha mümkün hale gelmektedir. Diğer yöntemlere kıyasla sahip olduğu avantajlar sayesinde, modern SLAM sistemleri çoğunlukla çizge tabanlı yöntem ile geliştirilmektedir.

### 3. İLGİLİ ÇALIŞMALAR

Bu bölümde literatürde yer alan, RGB-D kamera kullanılarak geliştirilen haritalama sistemleri incelenmiştir.

Bölüm 2.4.3'te de bahsedildiği üzere robotun yaptığı hareketin kestiriminde genel yaklaşım, kameradan gelen veri çerçevelerinin birbirleriyle karşılaştırılarak çerçeveler arasındaki dönüşümün hesaplanmasıdır. Literatürde yer alan haritalama yöntemleri, çerçeve karşılaştırma tekniklerine göre temel olarak iki ana grup şeklinde sınıflandırılabilir; nitelik tabanlı yöntemler ve yoğun yöntemler. Nitelik tabanlı yöntemler, çerçevelerden nitelik bulucular ile tespit edilen anahtar noktaların (*keypoint*) birbirleriyle eşleştirilmesi sonucunda elde edilen anahtar nokta çiftlerini kullanarak çerçeveleri karşıtırmaktadırlar. Yoğun yöntemler ise çerçevenin içerdiği verinin tamamını kullanmaya dayalı, piksel bazında hata minimizasyonunu benimseyerek robot hareketini hesaplamaktadırlar.

Diğer yandan haritanın genel tutarlılığı bağlamında, döngü kapama ve çizge optimizasyonu bileşenlerini içeren tam SLAM sistemlerinin yanı sıra, bu iki işlemlere sahip olmayan yani ortam haritasını sadece odometri tahmini vasıtasıyla çıkaran artırımsal (*incremental*) haritalama metotları da literatürde yer almaktadır.

İlerleyen alt kesimlerde, haritalama yöntemleri temel olarak kullandıkları çerçeve karşılaştırma tekniğine göre gruplandırılarak incelenmiştir.

#### 3.1. Nitelik Tabanlı Yöntemler

Nitelik tabanlı yöntemler çerçevedeki verinin tamamını kullanmak yerine, çerçeveden elde edilen anahtar noktalar ile işlem yaptıkları için yoğun yöntemlere göre çok daha düşük maliyetle çalışabilmektedirler. Genel yaklaşım; nitelik bulucu ile çerçevelerdeki anahtar noktaların tespit edilmesi, nitelik tanımlayıcı ile her bir nokta için o noktayı tanımlayan bir vektör oluşturulması, sonrasında ise tanımlayıcı vektörler karşılaştırılarak kaynak ve hedef çerçeveler arasındaki anahtar nokta eşleşmelerinin bulunmasıdır. Belirlenen eşleşmeler kullanılarak çerçeveler arasındaki dönüşüm hesaplanmakta ve robotun hareketi tespit edilmektedir.

RGB-D kamera kullanan ilk SLAM sistemlerinden birisi Henry ve diğerleri [23] tarafından geliştirilmiştir. Bu sistemde çerçeveden elde edilen anahtar noktalar geometrik karşılaştırma ile birlikte kullanılmaktadır. Kameradan gelen renkli

çerçevelerden SIFT [24] anahtar noktaları çıkarılmakta ve ilgili derinlik çerçeveleri kullanılarak anahtar noktaların 3B koordinatları tespit edilmektedir. RANSAC algoritmasının [25] nokta eşleşmeleri üzerinde uygulanmasıyla çerçeveler arasındaki dönüşüm hesaplanmaktadır. Sonrasında ise ICP algoritması [26] hesaplanan dönüşüm ile ilklendirilerek uygulanmakta ve dönüşümdeki hata azaltılmaktadır. Çizge oluşturulması aşamasında anahtar çerçeve (*keyframe*) mantığı kullanılmıştır. Anahtar çerçevelerin belirlenmesi görsel örtüşmeye göre gerçekleştirilmiştir. Herhangi bir çerçeve için çakıştırma işlemi gerçekleştirildikten sonra SIFT nitelikleri tekrar kullanılarak en son anahtar çerçeve ile arasındaki dönüşüm RANSAC ile hesaplanmaktadır. Elde edilen modele uygun değer (*inlier*) sayısı belirli bir eşik değer (*threshold*) altındaysa, anahtar çerçeve ile mevcut çerçeve arasında yeterli derecede benzerlik olmadığından yeni anahtar çerçeve eklenmektedir. Bu şekilde robot hareket ettikçe anahtar çerçeve kümesi genişlemektedir. Döngü kapama tespiti için; çizgeye eklenen her yeni anahtar çerçeveden sonra, önceki anahtar çerçeveler ile RANSAC dönüşüm hesaplaması gerçekleştirilmektedir. RANSAC sonucunda yeterli sayıda modele uygun değer elde edilmesi durumunda döngü kapama gerçekleştiğine karar verilmekte ve çizgede ilgili anahtar çerçevelere ait düğümler arasına yeni bir kenar eklenmektedir. Her döngü kapama tespitinden sonra, SLAM uygulamaları için geliştirilmiş bir optimizasyon yöntemi olan TORO [27] ile çizge üzerinde optimizasyon gerçekleştirilmektedir.

[28]'de Henry ve diğerleri, önceki yaklaşımları [23] üzerinde belirli değişiklikler ve iyileştirmeler yaparak daha gelişmiş bir SLAM sistemi sunmuşlardır. Önceki çalışmada kullanılan SIFT nitelikleri yerine FAST nitelik bulucusu [29] ile Calonder tanımlayıcısı [30] birlikte kullanılmıştır. Çerçeve çakıştırma işleminde; RANSAC sonucunda elde edilen modele uygun değer sayısı yeterli derecede ise ICP algoritması işletilmemekte, RANSAC ile bulunan dönüşüm kullanılmaktadır. Anahtar çerçeve belirlenmesi aşamasında, en son anahtar çerçeveden sonra geçilen konumlar göz önünde bulundurularak, toplam dönüş (*rotation*) ve öteleme (*translation*) belirli bir eşik değer (*threshold*) üzerindeyse yeni anahtar çerçeve eklenmektedir. Bu şekilde anahtar çerçeveler robot hareketine endeksli olarak belirlenmektedir. Döngü kapama tespitinde ise, bir kelime çantası (*bag of words*) yaklaşımı [31] kullanılarak anahtar çerçeveler filtrelenmekte ve RANSAC

algoritması seçilen çerçeveler için uygulanmaktadır. Çizge optimizasyonu ise Sparse Bundle Adjustment [32] ile yapılmaktadır.

Henry ve diğerlerinin çalışmalarına [23], [28] benzer bir yaklaşım Engelhard ve diğerleri [33] tarafından geliştirilmiştir. Renkli çerçeveden SURF [34] anahtar noktaları çıkarılıp önceki çerçevede bulunmuş olan niteliklerle eşleştirme yapıldıktan sonra bu noktaların pozisyonlarına ait derinlik verileri kullanılarak iki çerçeve arasındaki nokta tabanlı 3B ilişkiler kümesi elde edilmektedir. Bu ilişkiler taban alınarak, RANSAC algoritması kullanımıyla çerçeveler arasındaki dönüşüm tahmin edilmekte ve bu dönüşüm ICP kullanılarak geliştirilmektedir. Haritanın genel olarak tutarlı hale getirilmesi için ise Grisetti ve diğerlerinin yöntemi [35] kullanılarak konum çizgesi optimize edilmektedir.

Yine benzer bir SLAM sistemi Fioraio ve Konolige [36] tarafından önerilmiştir. Henry ve arkadaşlarının çalışmasından [23] farklı olarak, renk ve derinlik nitelikleri art arda gelen çerçevelerin çakıştırılması işleminin yanı sıra genel düzeltme (*adjustment*) işleminde de birlikte ele alınmıştır. Optimizasyon işlemi g2o yöntemi [37] ile gerçekleştirilmiştir. Sistemin bir başka özelliği ise, anahtar noktaların tespitinden önce ilgili derinlik çerçevelerinden alt örneklemeler alınarak gürültünün etkisinin azaltılması için filtreleme yapılmasıdır. Yine döngü kapama tespitinde çerçeveler arasındaki benzerlik kriteri olarak renkli görüntüdeki nitelikler yerine ICP algoritması ile tespit edilen nokta çiftlerinin sayısı kullanılmaktadır.

Huang ve diğerleri [38], görsel odometri için ilk önce renkli görüntüyü gri tonlamalı (*grayscale*) şekle dönüştürülerek *Gaussian kernel* ile düzleştirmektedir. Sonrasında ise Gauss piramidi oluşturularak piramidin her seviyesindeki nitelikler FAST nitelik bulucusu ile tespit edilmektedir. Her bir niteliğe denk gelen derinlik verisi de derinlik çerçevesinden elde edilmekte ve derinliğe sahip olmayan nitelikler elenmektedir. Art arda gelen çerçevelerdeki niteliklerin çoğunluğunun hareketinin 3B dönüşten kaynaklandığı varsayımıyla, bu dönüş tahmin edilerek çerçeveler arasında nitelik eşleştirmesi aşamasında arama alanı kısıtlanmaktadır. Mei ve diğerlerinin yöntemi [39] kullanılarak bu ilk dönüş hesaplanmakta ve sonrasında çerçeveler arasındaki nitelik eşleşmeleri, dönüş tahmini sonucu elde edilen arama alanında bulunmaları kriteri de kullanılarak belirlenmektedir. Howard'ın yaklaşımı [40] ile tutarlı nitelik eşleşmelerinin çizgesi oluşturulup açgözlü (*greedy*) algoritma kullanılarak aykırı değerler elenmektedir. Hareket

tahmini, eşleşen nitelikler üzerinden 3 adımda yapılmaktadır. İlk önce, Horn'un mutlak yönelim metodu [41] ile başlangıç tahmini yapılmakta, ardından nitelik *reprojection* hatası minimize edilmekte ve hareket tahmini iyileştirilmektedir. Son adımda ise, reprojection hatası belirli bir eşik değeri aşan nitelik eşleşmeleri modele uygun değerler kümesinden çıkarılmakta ve hareket tahmini tekrardan iyileştirilmektedir.

Endres ve diğerleri [42], renkli görüntü çerçevesinden anahtar noktaların elde edilmesi için SIFT, SURF ve ORB [43] yöntemlerini kullanmışlardır. Algılayıcıdan gelen her yeni çerçeve için; direkt önceki (*predecessor*) 3 çerçeve ve daha öncekiler arasından rastgele örneklenen 17 diğer çerçeve belirlenerek bu 20 çerçeve ile mevcut çerçeve arasında anahtar nokta eşleştirmesi yapılmaktadır. Sonrasında ise RANSAC ile dönüşüm hesaplanmakta ve yeterli modele uygun değer elde edilen dönüşümler için çizgeye yeni kenarlar eklenmektedir. Çizge optimizasyonu  $g_2o$  ile gerçekleştirilmektedir. Bu yöntemin genişletilmiş versiyonunda [44] ise, mevcut çerçeve ile arasında dönüşüm hesaplanacak olan çerçeveler 3 ana grup şeklinde seçilmektedir. Bu gruplar; direkt önceki çerçeveler, komşu çerçeveler ve anahtar çerçevelerdir. Bu 3 gruptan ayrı ayrı belirli sayıda çerçeve seçilerek mevcut çerçeve ile nitelik eşleştirmesinden sonra aradaki RANSAC dönüşümü hesaplanır. RANSAC sonrasında modele uygun değer sayısına göre seçilen dönüşümler üzerinde bir ortam ölçüm modeli (*environment measurement model*) kullanılarak doğrulama yapılır. Doğrulanmış dönüşümler için çizgede yeni kenarlar belirlenir. Burada ikinci gruptan seçilen çerçeveler daha yakındaki döngü kapamaların, üçüncü grup ise geniş döngü kapamaların tespit edilmesi için kullanılmaktadır. Ayrıca bu iki gruptan yapılan çerçeve seçimi rastgele gerçekleştirilmektedir.

Döngü kapama tespitini rastgele gerçekleştiren bir başka yöntem ise Maier ve diğerleri [45] tarafından önerilmiştir. Bu yöntemde dönüşüm hesaplanması Henry ve diğerlerinin yaklaşımına [28] benzer şekilde çerçevelerden elde edilen nitelikler üzerinde RANSAC algoritması uygulanmasıyla gerçekleştirilir. Döngü kapama tespiti için, geçmiş çerçevelerden örneklenen 20 çerçevenin her biri ile mevcut çerçeve arasında çakıştırma işlemi gerçekleştirilir. Çizgenin tamamının optimize edilmesi yerine, çizge alt haritalara (*submap*) bölünerek her bir parça ayrı ayrı

optimize edilir. Bu şekilde, optimizasyon problemi alt problemlere ayrılarak ele alınmaktadır.

Labbe ve Michaud [46]'un önerdiği RTAB-Map sistemi, döngü kapamaları tespit etmek için SURF anahtar noktalarını bir kelime çantası yaklaşımı içerisinde kullanmaktadır. Benzer şekilde döngü kapamaya ait çerçeveler arasında RANSAC ile anahtar nokta eşleşmeleri üzerinden dönüşüm hesaplanmakta ve modele uygun değer sayısına göre karar verilmektedir. Döngü kapamaların daha az maliyetle ve hızlı tespit edilebilmesi için geliştirilen bir bellek yönetim metodu kullanılmaktadır. Bu metotta bellek kısa süreli, çalışan ve uzun süreli olmak üzere üç parçaya ayrılmaktadır. Yeni eklenen düğümler kısa süreli bellekte tutulmakta, üzerinden zaman geçtikçe daha eski düğümler çalışan belleğe gönderilmektedir. Döngü kapama tespit edildiğinde ise uzun süreli bellekte yer alan komşu düğümler çalışan belleğe alınmaktadır. Çizge optimizasyonu TORO yöntemi ile gerçekleştirilmektedir.

Mur-Artal ve Tardos [47] tarafından geliştirilen ORB-SLAM2 sistemi, renkli görüntü çerçevelerinden ORB anahtar noktalarını elde etmekte ve haritalamaya dair tüm işlemleri bu noktalardan faydalanarak gerçekleştirmektedir. Sistemde kullanılan çizge anahtar çerçeve tabanlıdır. Anahtar çerçeveler, görüntüdeki anahtar noktaların derinlik verilerine göre yakın ya da uzak şeklinde sınıflandırılması yoluyla belirlenmektedir. İzlenen (*tracked*) yakın anahtar nokta sayısı yeterli olmadığında yeni anahtar çerçeve oluşturulmaktadır. Robot hareketinin tahmini için, 3B nokta eşleşmeleri arasındaki *reprojection* hatasının minimize edilmesine dayalı *bundle adjustment* yöntemi kullanılmaktadır. Döngü kapama tespiti için ise bir kelime çantası yaklaşımı [48] uygulanmaktadır. Her bir döngü kapama tespitinden sonra *full bundle adjustment* gerçekleştirilerek çizgedeki tüm anahtar çerçeveler üzerinde optimizasyon gerçekleştirilmektedir.

### **3.2. Yoğun Yöntemler**

Yoğun yöntemler, nitelik tabanlı yaklaşımlardaki gibi görüntüden anahtar nokta çıkarımı (*keypoint extraction*) yapmak yerine, çerçevedeki verinin tamamını kullanılmaktadırlar. Bu yöntemler hareket tahminini piksel bazındaki kısıtlar üzerinden ele almaktadırlar.

Steinbrücker ve diğerleri [49], odometri tahminini enerji tabanlı bir yaklaşımla gerçekleştirmeyi önermişlerdir. Ana fikir, döndürülmüş (*warped*) görüntülerin fototutarlılığını (*photo-consistency*) maksimize eden dönüşümün belirlenmesidir. Fototutarlılık teriminin temelinde yatan düşünce, ortamın durgun olduğu varsayımından dolayı her bir yüzey noktasının, görüldüğü tüm kamera görüntülerinde aynı renk değerine sahip olması gerekliliğidir. Görüntü çakıştırma aşamasında *backprojection* hatasının minimize edilmesi yoluna gidilmiştir. Enerji fonksiyonu olarak tanımlanan *backprojection* hatasının minimizasyonu için, bir dizi dış bükey optimizasyon problemi kullanılmıştır. Enerji doğrusallaştırılmış ve ortaya çıkan normal denklemi hareketi gösteren 3B dönüşüm koordinatları için çözülmüştür. Önerilen yaklaşım küçük çaplı hareketler için etkili sonuçlar üretebilmektedir.

[49]'daki çalışmanın genelleştirilmiş versiyonu Kerl ve diğerleri [50] tarafından sunulmuştur. Önerilen yöntem, çerçevelerin fotometrik hata minimizasyonu ile çakıştırılması prensibinin yanında, türetilen olasılıksal bir formülasyonun sisteme entegrasyonunu da kapsamaktadır. Bu formülasyon, algılayıcı ve hareket modelleri için özel olasılık dağılımlarının kullanımına olanak sağlamaktadır. Dolayısıyla hareket ön bilgisi sisteme dâhil edilerek daha başarılı hareket tahmini yapılabilmektedir. Bu yaklaşımın genişletilmiş tam SLAM versiyonu olan DVO SLAM [51], hareket tahmininde fotometrik hatanın yanında derinlik hatasını da minimize etmektedir. Sistemde anahtar çerçeve tabanlı bir yaklaşım kullanılmıştır. Anahtar çerçeve seçimi ve döngü kapama tespiti için entropi (*entropy*) tabanlı bir benzerlik ölçüsü geliştirilmiştir. Son anahtar çerçeve - mevcut çerçeve arasındaki hareket tahmini ile son anahtar çerçeve - bir sonraki çerçeve arasındaki hareketin entropi oranının belirli bir eşik değerden düşük olması durumunda, önceki çerçeve yeni bir anahtar çerçeve olarak seçilmekte ve haritaya eklenmektedir. Döngü kapama tespitinde ise aday çerçeveler, anahtar çerçeve konumu etrafında önceden belirlenmiş bir yarıçapa sahip küre içerisinde metrik en yakın komşu araması ile belirlenmektedir. Sonrasında her bir döngü kapama adayı ile mevcut çerçeve arasında dönüşüm hesaplanmaktadır. Mevcut çerçeve ile aralarındaki entropi oranı belirli bir eşik değerden yüksek olan adaylar için döngü kapamaya karar verilmekte ve çizgeye ilgili kenar eklenmektedir. Çizge optimizasyonu için g2o yöntemi kullanılmıştır. Veri kümesi işlendikten sonra her anahtar çerçeve için ek döngü kapama araması yapılmakta ve tüm çizge yeniden optimize edilmektedir.

Bir diğ er yoğ un yaklařımda ortam geometrisi, her bir vokselin (üç boyutlu piksel) yüzeye olan uzaklık bilgisini içeren İřaretli Uzaklık Fonksiyonu (*Signed Distance Function*, SDF) kullanılarak ifade edilmiřtir [52], [53]. Bu řekilde derinlik çerçevesi ile SDF arasında minimize edilebilecek bir hata fonksiyonu tanımlanarak robot konumu tahmin edilmektedir. Tanımlanan SDF bir voksel grid içerisinde tutulmakta ve uzaklıkları ile orantılı olarak; yüzeyin arkasında kalan vokseller için pozitif, üzerinde bulunanlar için sıfır ve algılayıcı ile yüzey arasındakiler için ise negatif deęerler üretmektedir. Modele eklenecek derinlik çerçevesinin hangi vokselleri içerdii tespit edilmekte, SDF'deki yüzey noktaları ile örtüşen vokseller arasındaki uzaklıđı minimize eden dönüşüm bulunarak robotun hareketi tahmin edilmektedir. Algoritma yinelemeli olarak çalışmaktadır. Önce mevcut SDF kullanılarak konum tahmini yapılmakta, ardından ise bulunan konum bilgisi ile SDF güncellenmektedir. Robot konum takibi için yalnızca derinlik verisi kullanılmakta, renk bilgisi ortam modelinin oluşturulması aşamasında işleme katılmaktadır. Küçük ortamlarda kullanılabilen sistem GPU üzerinde gerçekleştirilmiştir.

İřaretli Uzaklık Fonksiyonu türevi kullanan diğ er bir yaklařım, Newcombe ve diğ erleri [54] tarafından geliřtirilen KinectFusion sistemidir. Sahne gösterimi (*scene representation*) için Kesilmiş İřaretli Uzaklık Fonksiyonu (*Truncated Signed Distance Function*, TSDF) olarak adlandırılan hacimsel yapı kullanılmaktadır. Konum tespiti için ICP algoritması işletilmekte ve ortamda hareket edildikçe sürekli olarak hacimsel model güncellenmektedir. RGB-D algılayıcıdan elde edilen derinlik çerçevesi ile hacimsel modelden *raycasting* yoluyla elde edilen yüzey (*surface*) arasında ICP algoritması çalıştırılarak, gerçekleştirilen hareket tahmin edilmektedir. Bu řekilde hizalanan yeni çerçeve TSDF'ye entegre edilmekte ve TSDF güncellenmektedir. Bu artırımsal sistem GPU üzerinde çalışmakta ve küçük ortamlarda kullanılabilir. Döngü kapama tespiti yapılmadıđı için sürekli artan kayma hatası ve kullanılan hacimsel modelin çok fazla bellek gerektirmesi sistemin küçük ortamlarla kısıtlı řekilde çalışabilmesinin temel sebepleridir.

KinectFusion sisteminin genişletilmiş versiyonu olan Kintuous [55], TSDF'yi dinamik olarak hareket ettirmektedir. Robot hareketi ile birlikte hacimsel model de kaydırılmakta ve hareket belirli bir eşik deęerden büyük olduđunda ortamın yeni bir bölgesi TSDF'ye giriş yapmaktadır. Hacimsel modelden çıkarılan bölge



kullanılarak bir nokta bulutu (*point cloud*) oluşturulmakta ve ortamın *mesh* gösterimine entegre edilmektedir. Bu şekilde TSDF robotun gezinimiyle birlikte hareket ettirilerek ortam haritası artırımsal olarak oluşturulmaktadır. Odometri tahmini için sadece derinlik verisinin kullanılması geometrik yapısı yetersiz ortamlarda, çerçeveler arasında gerçekleşen robot hareketleri büyük ölçekli olduğunda ya da algılayıcının kısıtlı menzilinden dolayı derinlik verisi bazı sahnelerde eksik olduğunda problemlere sebebiyet vermektedir. Bu problemlerin üstesinden gelenebilmesi için [56]'da Kintinuous sistemine görsel odometri bileşeni eklenmiştir. Steinbrücker ve diğerlerinin [49] RGB-D görsel odometri metodu GPU üzerinde gerçekleştirilmiş ve ICP ile birleştirilerek Kintinuous'a entegre edilmiştir. Bu şekilde, yalnızca derinlik verisi kullanımı dezavantajlarının aşılması ve odometri tahmininin daha başarılı yapılması hedeflenmiştir.

Whelan ve diğerleri [57], döngü kapama tespiti ve çizge optimizasyonu yeteneklerini [58] ekleyerek Kintinuous'ın artırımsal metodunu tam SLAM sistemi olarak genişletmişlerdir. Sistemde odometri tahmini yoğun bir yaklaşımla yapılmasına rağmen, döngü kapama tespiti için nitelik tabanlı bir metot benimsenmiştir. Çerçeveden SURF anahtar noktaları çıkarılmakta ve döngü kapama adayları bir kelime çantası veri tabanında aranmaktadır. Mevcut çerçeve ile eşleşen adaylar için 3 farklı eşik değeri uygulanarak doğrulama yapılmaktadır. Bu eşik değerleri; çerçeveler arasındaki anahtar nokta eşleşme sayısı, RANSAC sonucunda elde edilen modele uygun değeri sayısı ve ICP algoritmasının ürettiği ortalama hatadır. Ayrıca hacimsel yapının daha etkin kullanılabilmesi için TSDF algılayıcı pozisyonuna göre dinamik şekilde konumlandırılmaktadır. Bunun yanında, optimizasyonun daha az sayıda düğümle gerçekleştirilerek işlem zamanının düşürülmesi için çizgede alt örnekleme (*subsampling*) uygulanmaktadır.

Stückler ve Behnke [59] tarafından geliştirilen MRSSMap, hareket tahmini için RGB-D çerçevelerden elde edilen çoklu-çözünürlüklü (*multi-resolution*) surfel haritalarını çakıştırmaktadır. Çakıştırma işleminde öncelikle surfel ilişkilendirmesi yapılmakta, daha sonra ise eşleşme olasılığını (*matching likelihood*) maksimize eden dönüşüm tahmin edilmektedir. Sistemde anahtar görüşümlerden (*view*) oluşan bir çizge kullanılmaktadır. Çizge artırımsal şekilde, robot hareketi belirli bir eşik değerden büyük olduğunda yeni bir konum eklenerek genişletilmektedir.

Döngü kapama adayları rastgele belirlenmekte ve benzerlik ölçütü olarak anahtar görünüşler arasındaki eşleşme olasılığı kullanılmaktadır. Çizge g2o kullanılarak optimize edilmektedir.

ElasticFusion [60], bozulma (*deformation*) çizgesi kullanarak oda tarzı küçük boyuttaki ortamların *surfel* tabanlı modellerini oluşturabilmektedir. Sistem, modeldeki daha yakın zamanda gözlemlenmiş alanları aktif olarak, daha eski surfelleri ise pasif olarak etiketlemektedir. Mevcut çerçeve aktif alanla çakıştırılarak hem robot konumunun takibi sağlanmakta hem de yeni veriler modele entegre edilmektedir. Yerel (daha yakın) döngü kapamaların tespiti için, mevcut çerçeveye ait aktif alan, aynı çerçeveye yakın pasif alanla çakıştırılmaya çalışılmaktadır. Çakıştırma işlemi başarılı olursa, gerçekleşen yerel döngü kapama neticesinde ilgili pasif alan tekrar aktif olarak etiketlenir. Daha geniş döngü kapamaların tespiti için ise bir rastgele *fern encoding* yaklaşımı [61] kullanılmaktadır. Geniş döngü kapama tespit edilmesi durumunda surfel tabanlı haritada genel bir hizalama işlemi gerçekleştirilmektedir.

### 3.3. Diğer Yöntemler

Literatürde, konuyu diğer genel yöntemlerden daha farklı şekilde ele alan çalışmalar da yer almaktadır. Bu çalışmalara örnek olarak; kullanıcı etkileşiminden faydalanılması, yüksek çözünürlüklü görüntüler elde edilmesi, farklı odometri yaklaşımları arasında geçiş yapılması ya da panoramik çerçeveler oluşturulması verilebilir.

Du ve diğerleri [62], kapalı ortamların 3B haritalarının çıkarılması için çevrimiçi kullanıcı etkileşimine dayalı bir model geliştirmiştir. Çalışmada, Henry ve diğerlerinin sunduğu sistem [23] üzerine kullanıcı etkileşimi entegre edilmiştir. Kullanıcı algılayıcıyı serbestçe hareket ettirerek ortamı tarayabilmekte, sistem çevrimiçi geri bildirim sağlayarak kullanıcıyı yönlendirmektedir. Çerçeve çakıştırmasının başarısız olması durumunda (örneğin karanlık ya da dokusuz ortamlarda) sistem kullanıcıyı uyarmakta ve haritalama işlemi bekletilmektedir. Kullanıcı sistemden aldığı öneriler doğrultusunda kamerayı yeniden yönlendirmekte ve çakıştırma başarılı olduğunda haritalama işlemi kaldığı yerden devam etmektedir. Diğer yandan kullanıcı döngü kapama tespiti üzerinde de kontrole sahiptir. Sistemde 3B haritanın gerçek zamanlı olarak nasıl büyüdüğü

gözlemlenebilmekte, bakış açısı istenilen şekilde değiştirilerek tutarsızlıklar ve haritanın ne kadar tamamlandığı görülebilmektedir. Mevcut haritanın tamlığı devamlı olarak kontrol edilmekte, tamamlanmayan kısımlar hakkında görsel geri bildirim sağlanmaktadır. Bu şekilde daha doğru, detaylı ve tam haritalar elde edilmesi amaçlanmıştır.

Meilland ve Comport [63], hem fotometrik hem de derinlik hatalarının minimize edilmesiyle konum tahmini gerçekleştirmektedirler. Bir ters birleşimsel model içerisinde düşük çözünürlüklü girdi görüntüleri birleştirilmekte ve yüksek çözünürlüklü görüntüler oluşturulmaktadır. Bu şekilde yüksek çözünürlüklü bir harita oluşturulmaktadır. Sistem GPU üzerinde gerçekleştirilmiştir. Ayrıca [64]'te, hacimsel ve anahtar çerçeve tabanlı yaklaşımların bütünleştirilmesi önerilmiştir. Kapanmaların (*occlusion*) ele alınması için voksel tabanlı modelin ve anahtar çerçeve gösteriminin temel avantajlarından faydalanılmıştır. Birden fazla çerçeve birleştirilerek genişletilmiş anahtar çerçeveler oluşturulmakta ve bu şekilde çakıştırma performansı arttırılmaktadır.

Hu ve diğerleri [65], RGB-D ve tek kamera tabanlı teknikler arasında geçiş sağlayan bir SLAM sistemi geliştirmişlerdir. Önerilen yöntem, derinlik verisi yeterli olmadığına ya da ortamın geometrik yapısı yetersiz olduğunda tek kamera tabanlı tekniği kullanmaktadır. RGB-D ve tek kamera tekniklerinin oluşturdukları farklı yerel haritalar birleştirilerek ortamın genel haritası elde edilmektedir. Döngü kapama tespiti için anahtar çerçeve eşleşmelerinden faydalanılmaktadır.

Taylor ve diğerleri [66], haritalama işleminde derinlik panoramalarının kullanımını önermektedirler. Art arda gelen derinlik çerçeveleri ICP algoritması ile hizalanarak 360° panoramik derinlik anahtar çerçeveleri oluşturulmaktadır. Bu panoramik anahtar çerçeveler ICP kullanımıyla çakıştırılarak ortamın 3B modeli elde edilmektedir.

### **3.4. Veri Kümeleri**

SLAM sistemlerinin başarımlarının değerlendirilmesi için yaygın olarak kullanılan RGB-D veri kümeleri ICL-NUIM [67] ve TUM RGB-D Benchmark [68] veri kümeleridir.

### 3.4.1. ICL-NUIM Veri Kümesi

Handa ve diğerleri [67] tarafından oluşturulan veri kümesi, oda büyüklüğünde iki farklı yapay ortama (*Living Room* ve *Office Room*) ait sahneler içeren RGB-D görüntü kayıtları içermektedir. Bu yapay ortamlarda, gerçek dünyaya benzer aydınlatma koşulları simüle edilmiştir (Bkz. Şekil 3.1). Üretilen senaryolarda görüntü kayıtları, elde taşınan bir RGB-D algılayıcı ile ortamda dolaşılmasıyla oluşan çeşitli yolları içermektedir. Bu yollara ait kesin referans (*ground truth*) konum verileri de veri kümesi tarafından sağlanmaktadır. Ortamda yapılan yollar kısadır ve hemen hemen hiç döngü kapama içermemektedir. Dolayısıyla bu veri kümesinde yer alan görüntü kayıtları için SLAM sisteminin başarısı büyük oranda odometri tahmininin performansına bağlıdır. Veri kümesinde, algılayıcı gürültüsünün mevcut olmadığı senaryolarla üretilen görüntü kayıtlarının yanı sıra, aynı kayıtların gürültülü versiyonları da kullanıma sunulmuşlardır. Bunun için hem RGB hem de derinlik çerçevelerine simüle edilmiş gerçekçi algılayıcı gürültüsü eklenmiştir.



*Living Room*



*Office Room*

Şekil 3.1. ICL-NUIM veri kümesinden çeşitli sahneler

### 3.4.2. TUM RGB-D Benchmark Veri Kümesi

Sturm ve diğerlerinin [68] oluşturduğu TUM RGB-D Benchmark veri kümesi, RGB-D algılayıcıyla orta boyutlu ve geniş olmak üzere iki farklı ortamda kaydedilen çeşitli görüntü kayıtlarından oluşmaktadır. Her bir kayıt için; RGB ve derinlik çerçevelerinin yanında, yapılan yol için hareket yakalama (*motion capture*) sistemiyle elde edilen kesin referans verileri de bulunmaktadır. Veri kümesi içerisinde çok çeşitli sahneler ile algılayıcı hareketleri mevcuttur. Görüntü kayıtları, algılayıcının hareketine ve kaydın yapıldığı ortama bağlı olarak değişik türlerde zorluklar içermektedirler.

*fr1* kayıtları, dokulu sahneler içeren orta ölçekli bir ofiste kaydedilmişlerdir (Bkz. Şekil 3.2). Dolayısıyla bu kayıtlar içerdikleri anahtar nokta sayısı bakımından zenginlerdir. Kayıtlarda yapılan yolların uzunluğu, birkaç daha kısa istisna dışında yaklaşık 10 m - 15 m aralığındadır. Bu gruptaki kayıtlar nitelik açısından zengin olmalarına rağmen, hareket bulanıklığı, ortam ışık seviyesindeki değişimler ve hızlı algılayıcı hareketleri gibi zorluklar içermektedirler. Ayrıca bazı kayıtlarda küçük çaplı döngü kapamalar yer almaktadır.



*fr1*



*fr2*

Şekil 3.2. TUM RGB-D Benchmark veri kümesinden çeşitli sahneler

*fr2* kayıtları, büyük bir endüstriyel salonda kaydedilmişlerdir. Bu kayıtlar, *fr1* grubundakilerden daha zor şartlara sahip sahneler içermektedirler. Bu kayıtlarda

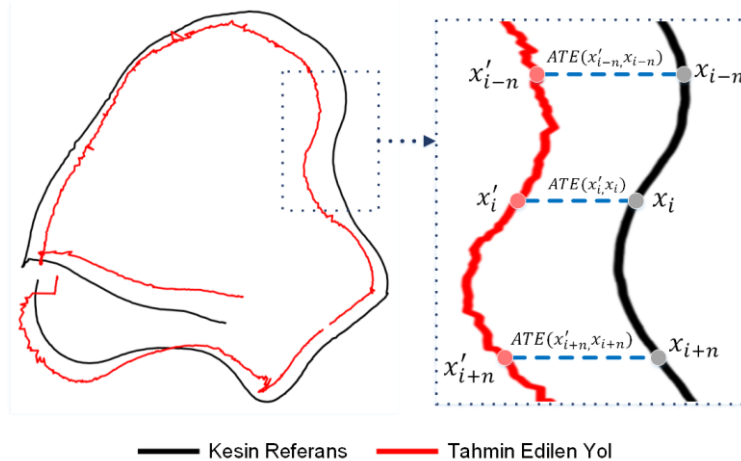
aydınlatmadaki deęişimler, hareket bulanıklığı, hızlı hareketlerin yanında; dokusuz ortam dolayısıyla düşük sayıda anahtar nokta, kayıt içerisinde kesintiler, ortamdaki ince yapılardan ya da algılayıcının menzil kısıtından kaynaklı yanlış ya da eksik derinlik verileri ve tekrarlı yapılar gibi daha çetin şartlar söz konusudur. Bu grup, uzun yollar ve geniş döngü kapamalar içeren kayıtlara sahiptir.

### 3.5. Ölçüm Metrięi

SLAM sistemlerinin başarımlarını ölçümünde yaygın olarak kullanılan metrik mutlak yol hatasıdır (*absolute trajectory error*, ATE) [68]. Oluşturulan haritanın başarısı, robotun bulunduğu konumların ne kadar doğru tahmin edildięi ile direkt ilişkilidir. Bu hata metrięi, robotun ortamda yapmış olduęu yol için SLAM sisteminin yaptığı tahmin ile kesin referans arasındaki kayma miktarını ifade etmektedir. Bunun için, tahmin edilen yol ile kesin referans arasında konum bazında ilişkilendirme yapılarak aradaki mesafe hesaplanır (Bkz. Şekil 3.3). Kullanılan her iki veri kümesi için de önerildięi üzere yol kayması, ATE'nin kare ortalamalarının karekökü (*root mean square*) şeklinde aşığıdaki gibi ifade edilir;

$$RMS - ATE(X', X) = \sqrt{\frac{1}{N} \sum_{i \in [N]} \|trans(x'_i) - trans(x_i)\|^2} \quad (3.1)$$

Burada  $X'$  tahmin edilen robot yolunu  $\{x'_1, \dots, x'_N\}$ ,  $X$  ise robot yoluna dair kesin referansı ifade eder.  $trans(x'_i)$  ifadesi, tahmin edilen yoldaki  $x'_i$  konumunun 3B öteleme koordinatlarını (x,y,z eksenlerinde) simgeler. Ayrıca  $\{1, 2, 3, \dots, N\}$  kümesi  $[N]$  ile gösterilmiştir. Bu şekilde, tahmin edilen her bir robot konumu ile kesin referanstaki ilgili konum ilişkilendirilerek aralarındaki *Euclidean* uzaklığının kare ortalamalarının karekökü hesaplanır.



Şekil 3.3. Mutlak yol hatasının hesaplanması

## 4. RGB-D SLAM METOTLARINDAKİ NİTELİK BULUCU VE TANIMLAYICILARIN KARŞILAŞTIRILMASI

Nitelik tabanlı çerçeve çakıştırma tekniği, aralarında mesafe bulunan yani art arda olmayan veri çerçeveleri için yoğun yöntemlere göre daha başarılı hareket tahmini sağlamaktadır. Yoğun yöntemler ise odometri tahmininde daha başarılıdır çünkü art arda gelen çerçeveler arasında daha iyi dönüşüm hesaplayabilmektedirler. İşlem yükü açısından bakıldığında ise; çerçevelerden anahtar nokta çıkarımı yaparak dönüşüm hesaplayan nitelik tabanlı yöntemler, yoğun yöntemlere göre çok daha az miktarda veri ile çalıştıkları için daha hızlıdır. Yoğun yöntemler çerçevelerdeki tüm veriyi işlediklerinden dolayı yüksek işlem gücü gerektirmekte ve daha yavaş çalışmaktadırlar.

SLAM kapsamında döngü kapamaların doğru tespiti çok önemlidir ve döngüyü kapayan çerçeveler arasında büyük mesafeler olabilmektedir. Bu sebeple, hem aralarında mesafe bulunan çerçeveler arasında daha başarılı dönüşüm hesaplayabilmesiyle hem de daha az işlemsel yük gerektirmesiyle nitelik tabanlı yaklaşım çok küçük ortamlar dışında daha avantajlıdır.

Nitelik tabanlı çalışan bir SLAM sistemi için, kullanılan nitelik bulucu ve tanımlayıcının sistem performansı üzerinde önemli etkisi bulunmaktadır. Bu bölümde, çeşitli nitelik bulucu ve tanımlayıcıların bir RGB-D SLAM sisteminin genel başarımı ve hızı üzerine etkileri incelenmiştir [69].

### 4.1. Kullanılan Nitelik Bulucu ve Tanımlayıcılar

Nitelik bulucular, çerçevedeki dikkat çekici bölgelerde yer alan belirgin anahtar noktaları (*keypoint / feature point / interest point*) tespit ederler. Nitelik tanımlayıcılar ise, her bir anahtar noktanın etrafındaki bölgeyi kullanarak noktayı tanımlayan bir vektör oluştururlar. Tanımlayıcı vektörler, her bir anahtar noktanın diğerlerinden ayrılması için gerekli bilgileri içermektedirler. Nitelik tanımlayıcılar, oluşturdukları vektörlerin içerdikleri veri türüne göre kayan noktalı (*floating point*) ya da ikili (*binary*) tanımlayıcılar şeklinde sınıflandırılabilirler.

Çerçevelerin çakıştırılması için, ilk olarak içerdikleri anahtar noktalar tespit edilir ve her bir nokta için tanımlayıcı vektör oluşturulur. Daha sonra, kaynak ve hedef çerçevedeki noktalara ait tanımlayıcı vektörler arasındaki uzaklığın büyüklüğüne göre, kaynak çerçeve üzerindeki her bir anahtar noktaya, hedef çerçeve üzerinde

karşılık gelen bir nokta eşleştirilmektedir. Sonrasında ise, eşleşen noktalar kullanılarak çerçeveler arasındaki dönüşüm hesaplanmaktadır [70]. Anahtar noktaların belirlenmesi ve eşleştirilmesi işlemi, hesaplanacak dönüşümün doğruluğunu önemli ölçüde etkilemektedir. Bu sebeple, yanlış eşleşmelerin sayısı ne kadar azaltılabilirse, hesaplanacak dönüşümdeki hata oranı bir o kadar düşecektir.

Bu bağlamda, nitelik tabanlı bir RGB-D SLAM sistemi üzerinde farklı nitelik bulucu ve tanımlayıcı kombinasyonları kullanılarak, hem oluşturulan haritanın doğruluğu hem de sistem hızı açısından performans değerlendirmesi yapılmıştır.

Kullanılan nitelik bulucular aşağıdaki gibidir;

- SIFT [24]
- SURF [34]
- BRISK [71]
- ORB [43]
- FAST [29]
- GFTT (*Good Features to Track*) [72]
- CenSurE [73]

Kullanılan nitelik tanımlayıcılar ise;

- SIFT
- SURF
- BRISK
- ORB
- BRIEF [74]
- FREAK [75]

şeklindedir.

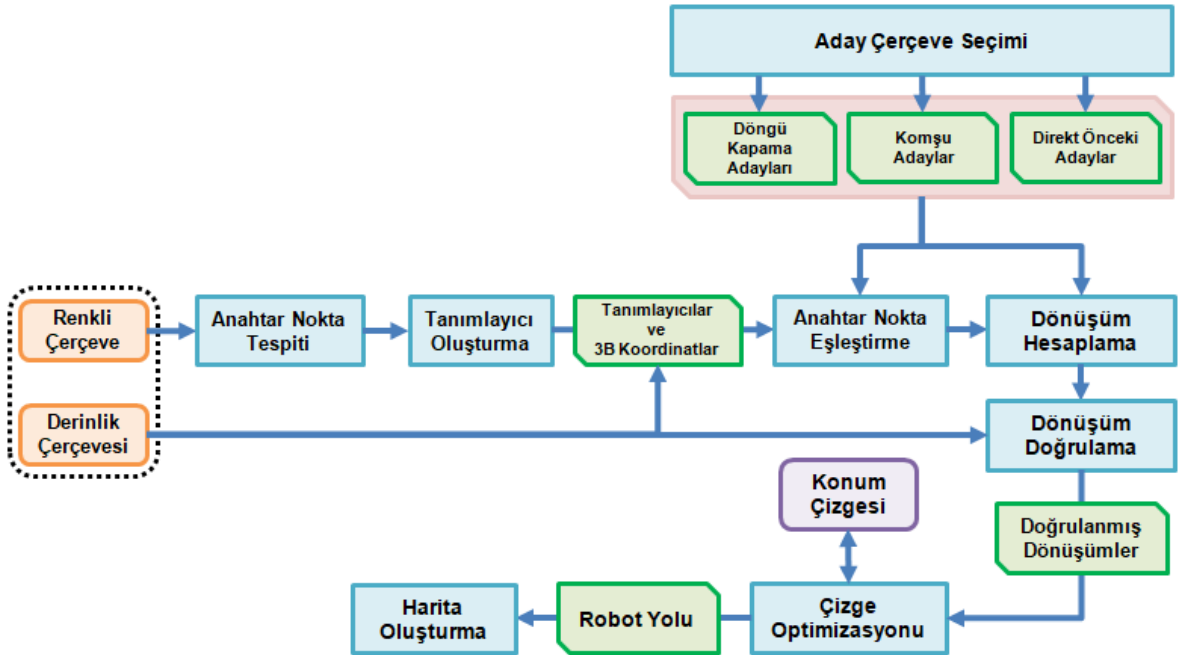
## **4.2. RGB-D SLAM Sistemi**

Bu çalışmada literatürdeki en başarılı yaklaşımlardan birisi olan, Endres ve diğerlerinin [44] RGB-D SLAM sistemi kullanılmıştır. Sistemin genel işleyişi Şekil 4.1'de görülmektedir.

Sistemde çerçeveler arasındaki robot hareketi tahmini nitelik tabanlı bir yöntemle gerçekleştirilmektedir. Algılayıcıdan gelen her bir RGB çerçeve üzerinde nitelik



bulucu kullanımıyla anahtar noktalar tespit edilmektedir. Sonrasında ise nitelik tanımlayıcı vasıtasıyla, tespit edilen noktalara ait tanımlayıcı vektörler elde edilmektedir. Bir sonraki adımda, her bir nokta için ilgili derinlik çerçevesinde yer alan derinlik verisi kullanılarak noktaya ait 3B koordinatlar hesaplanmaktadır. Çakıştırılacak çerçeveler arasında anahtar nokta eşleştirmesi yapılarak, bulunan eşleşmelerin 3B koordinatları üzerinden RANSAC kullanımıyla çerçeveler arası dönüşüm hesaplanmaktadır. RANSAC algoritmasındaki her bir yinelemede; bir en küçük kareler yöntemi [76] kullanılarak eşleşen nokta çiftleri arasındaki dönüşüm hesaplanmakta ve kaynak-hedef noktalar arasındaki *Mahalanobis* uzaklığına göre modele uygun değerler belirlenmektedir. RANSAC algoritması işletildikten sonra, modele uygun değer sayısı belirli bir eşik değerden fazla ise bir ortam ölçüm modeli kullanılarak hesaplanan dönüşüm için doğrulama yapılmaktadır. Dönüşüm doğrulanabilirse, çerçeveler arasında başarılı bir dönüşüm hesaplandığı varsayımıyla, robotun ilgili aralıkta yapmış olduğu hareket tahmin edilmiş olmaktadır.



Şekil 4.1. RGB-D SLAM sisteminin [44] genel işleyişi

Algılayıcıdan gelen her bir yeni çerçeve için çizgeye yeni bir düğüm eklenmektedir. Eklenen her yeni düğüm için (robotun yeni konumu için), daha önce işlenmiş olan çerçeveler içerisinde seçilen adaylar ile dönüşüm hesaplaması yapılmaktadır.

Aday çerçeveler 3 farklı gruba ayrılmaktadır;

- Direkt önceki adaylar; mevcut çerçeveden direkt önceki  $n$  çerçevedir.
- Komşu adaylar; bir önceki çerçevenin çizgedeki komşuluğu içerisinde rastgele belirlenen  $k$  adet çerçeveden oluşmaktadır.
- Döngü kapama adayları; anahtar çerçeveler içerisinde seçilen  $l$  adet çerçeveden oluşmaktadır. Anahtar çerçeveler kümesi, mevcut çerçevenin en son anahtar çerçeve ile arasında başarılı bir dönüşüm hesaplanamaması durumunda yeni anahtar çerçeve olarak kümeye eklemesi yoluyla belirlenmektedir.

Mevcut çerçeve ile aday çerçeveler arasında başarılı şekilde hesaplanabilen dönüşümler için çizgede ilgili düğümler arasında kenar oluşturulmaktadır. Bu yöntemle, algılayıcıdan veri geldikçe front-end tarafından yeni düğüm ve kenarların eklenmesiyle çizge büyümektedir. Back-end çizgeyi g2o yöntemini kullanarak optimize etmekte, ayrıca çizge üzerinde hata değeri belirli bir seviyeden yüksek olan kenarları budamaktadır. Bu şekilde robotun hareketi süresince bulunduğu konumlar hesaplanmakta, dolayısı ile izlemiş olduğu yol tespit edilmektedir. Her bir konuma ait ölçüm verilerinin ortak bir koordinat sistemi üzerinde birleştirilmesi ile ortamın haritası elde edilmektedir.

### 4.3. Deneysel Kurulum

Gerçekleştirilen deneylerde TUM RGB-D Benchmark veri kümesindeki *fr1* kayıtları kullanılmıştır. Bu kayıtlar dokulu sahneler içermeleri sebebiyle anahtar nokta sayısı bakımından zenginlerdir. Dolayısıyla nitelik çıkarımı ve eşleştirme başarımının ölçülmesi için ideallerdir. Kullanılan kayıtlar hakkında ayrıntılı bilgiler Çizelge 4.1'de sunulmuştur. Deneyler, üzerinde Ubuntu 12.04 işletim sistemi çalışan, Intel Core i7-2600 CPU 3.40GHz işlemci, 8GB RAM ve NVIDIA GeForce GTX 550 Ti grafik kartı bulduran bir bilgisayarda gerçekleştirilmiştir. Deneylerde nitelik bulucu ve tanımlayıcıların *OpenCV* gerçekleştirmeleri [77] kullanılmıştır.

Çizelge 4.1. Deneylerde kullanılan *fr1* görüntü kayıtları hakkında ayrıntılı bilgiler

Kayıt	Süre (s)	Yol Uzunluğu (m)	Ort. Öteleme Hızı (m/s)
fr1/360	28.7	5.82	0.210
fr1/desk	23.4	9.26	0.413
fr1/desk2	24.9	10.16	0.426
fr1/floor	49.9	12.57	0.258
fr1/plant	41.5	14.80	0.365
fr1/room	48.9	15.99	0.334
fr1/rpy	27.7	1.66	0.062
fr1/teddy	50.8	15.71	0.315
fr1/xyz	30.1	7.11	0.244

Çizelge 4.2'de görüldüğü üzere deneylerde, bir çerçeveden çıkarılan maksimum anahtar nokta sayısı 700 olarak belirlenmiştir. Dönüşüm hesaplanması sırasında RANSAC algoritması için yineleme sayısı 250 olarak seçilmiştir. Bu parametre değerleri, çeşitli deneyler gerçekleştirilerek elde edilen sonuçlar neticesinde belirlenmiştir. Aday çerçeve sayıları için  $n$ ,  $k$  ve  $l$  parametrelerine eşit değerler uygulanmış olup, 4, 8 ve 12 değerleri denenmiştir. Yani aday değerinin 4 olması  $n = k = l = 4$  anlamına gelmektedir. Optimizasyon işlemi tüm veri çerçeveleri işlendikten sonra gerçekleştirilmiştir.

Çizelge 4.2. Deneylerde uygulanan parametre değerleri

Parametre	Değer
<b>Dönüşüm Parametreleri</b>	
Maksimum RANSAC yineleme sayısı	250
Maksimum anahtar nokta sayısı	700
<b>Aday Parametreleri</b>	
Direkt önceki aday sayısı ( $n$ )	4,8,12
Komşu aday sayısı ( $k$ )	4,8,12
Döngü kapama adayı sayısı ( $l$ )	4,8,12

Nitelik tanımlayıcı olarak BRIEF ve FREAK tüm nitelik bulucularla birlikte denenmiştir. Kendi tanımlayıcılarına sahip olan SIFT, SURF, BRISK ve ORB için, bu tanımlayıcılar kendi nitelik bulucuları ile birlikte test edilmiştir. Ayrıca, GFTT içerisinde Harris bulucusunun [78] aktif olduğu (GFTT\_HARRIS şeklinde ifade edilmiştir) durum da denenmiştir. Nitelik eşleştirme işleminde, ikili nitelik tanımlayıcıları (BRISK, ORB, BRIEF ve FREAK) için *Bruteforce-Hamming*

yöntemi, kayan noktalı tanımlayıcılar (SIFT ve SURF) için ise *FLANN* yöntemi kullanılmıştır.

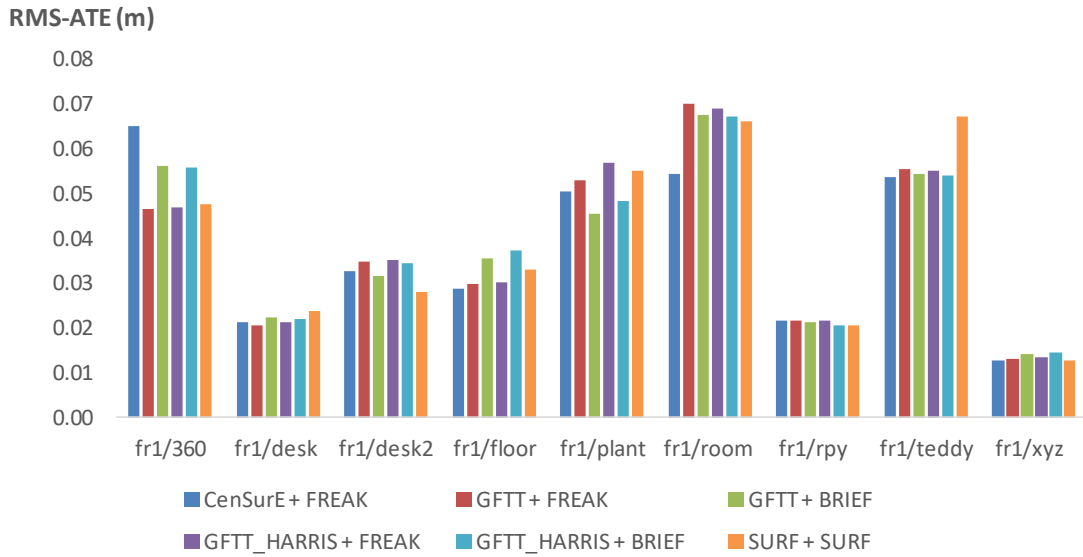
#### 4.4. Sonuçlar

Çizelge 4.3'te *fr1* görüntü kayıtları için elde edilen mutlak yol hatası sonuçları listelenmektedir. Çizelgedeki sonuçlar açısından, daha düşük RMS-ATE değeri daha doğru robot yolu tahmini anlamına gelmektedir. Ortalamaya bakıldığında, 3.79 cm hata payı ile en başarılı sonuçlar CenSurE nitelik bulucusu ile FREAK nitelik tanımlayıcısının birlikte kullanımı ile elde edilmiştir. Aday parametresinin 4 ve 8 değerleri için en başarılı kombinasyon yine CenSurE+FREAK, 12 değeri için ise SIFT+SIFT'tir. Yine ortalama sonuçlara göre, GFTT nitelik bulucusu hem BRIEF hem de FREAK tanımlayıcısı için CenSurE+FREAK'e yakın performans sergilemiştir.

Çizelge 4.3. *fr1* kayıtları için mutlak yol hatası (RMS-ATE) sonuçları (m)

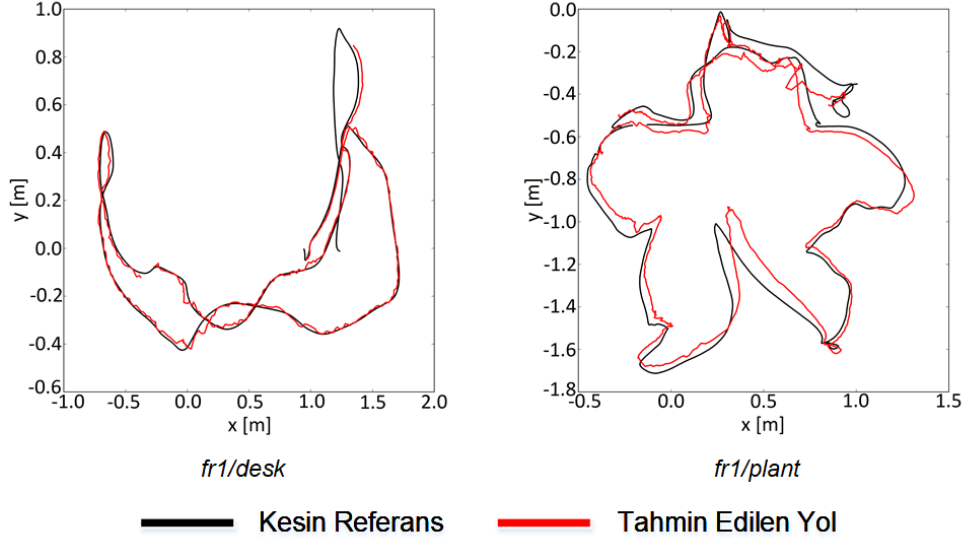
Nitelik Bulucu	Nitelik Tanımlayıcı	Aday Çerçeve Sayısı			Ortalama
		4	8	12	
BRISK	BRIEF	0.0461	0.0491	0.0381	0.0444
BRISK	FREAK	0.0559	0.0430	0.0411	0.0467
BRISK	BRISK	0.0502	0.0464	0.0460	0.0475
FAST	BRIEF	0.0680	0.0587	0.0567	0.0611
FAST	FREAK	0.0832	0.0629	0.0584	0.0682
GFTT	BRIEF	0.0402	0.0377	0.0383	0.0387
GFTT	FREAK	0.0396	0.0380	0.0373	0.0383
GFTT_HARRIS	BRIEF	0.0404	0.0390	0.0387	0.0394
GFTT_HARRIS	FREAK	0.0399	0.0392	0.0376	0.0389
ORB	BRIEF	0.1081	0.1134	0.0824	0.1013
ORB	FREAK	0.2614	0.2181	0.2329	0.2375
ORB	ORB	0.1630	0.1601	0.1058	0.1430
SIFT	BRIEF	0.0551	0.0521	0.0515	0.0529
SIFT	FREAK	0.0459	0.0472	0.0436	0.0456
SIFT	SIFT	0.0426	0.0387	<b>0.0372</b>	0.0395
CenSurE	BRIEF	0.0460	0.0383	0.0381	0.0408
CenSurE	FREAK	<b>0.0395</b>	<b>0.0365</b>	0.0377	<b>0.0379</b>
SURF	BRIEF	0.0490	0.0405	0.0396	0.0430
SURF	FREAK	0.0526	0.0492	0.0497	0.0505
SURF	SURF	0.0409	0.0389	0.0385	0.0394
Ortalama		0.0684	0.0624	0.0575	0.0627

Şekil 4.2'de, en başarılı sonuçları üreten 6 kombinasyon ile *fr1*'de yer alan kayıtlar için elde edilen ortalama hata oranlarının dağılımı görülmektedir. Ortam içerisinde döngü kapama gerçekleştirilen *fr1/360*, *fr1/floor*, *fr1/plant*, *fr1/room* ve *fr1/teddy* kayıtlarından *fr1/floor* haricindekilerde hata oranları genel olarak daha yüksektir. *fr1/floor* kaydında ise algılayıcının daha dengeli hareket ettirilmesi dönüşüm hesaplama başarımını arttırmış, dolayısıyla kayma hatasının nispeten daha düşük olmasını sağlamıştır.



Şekil 4.2. En başarılı 6 yöntemle ait, *fr1* kayıtları için ayrıntılı hata oranları (Aday parametresinin 4, 8 ve 12 değerleri için elde edilen sonuçların ortalaması)

Şekil 4.3'te, CenSurE+FREAK kullanımı ve aday parametresinin 8 değeri için *fr1/desk* ve *fr1/plant* kayıtlarındaki robot yolu tahminleriyle kesin referanslar karşılaştırılmıştır. *fr1/desk* kaydı için elde edilen sonucun kesin referansa çok yakın olduğu gözlemlenmektedir. Diğer yandan, *fr1/plant* için tahmin edilen yol daha yüksek kayma hatası içerdiğinden kesin referans ile arasındaki fark daha belirgindir.



Şekil 4.3. *fr1/desk* ve *fr1/plant* kayıtları için robot yoluna ait kesin referansın ve CenSurE+FREAK ile tahmin edilen sonucun x-y düzlemine izdüşümü (Aday = 8)

SLAM sistemleri için yeterince doğru sonuçlar üretmek kadar hızlı çalışmak da önemlidir. Çizelge 4.4'te sistemin çalışma hızına dair sonuçlar görülmektedir. Çizelgede yer alan işlem süreleri hesaplanırken optimizasyon için harcanan zaman da hesaba katılmıştır. Ortalama olarak en hızlı yöntem çerçeve başına 86.6 milisaniye işlem zamanı ile ORB+FREAK kombinasyonudur. Gerçekleştirilen deneylerde, ORB nitelik bulucusu birlikte kullanıldığı tüm tanımlayıcılar ile hız açısından en başarılı sonuçları üretmesine rağmen, Çizelge 4.3'te de görüldüğü üzere doğruluk oranı açısından tersi durum söz konusudur.

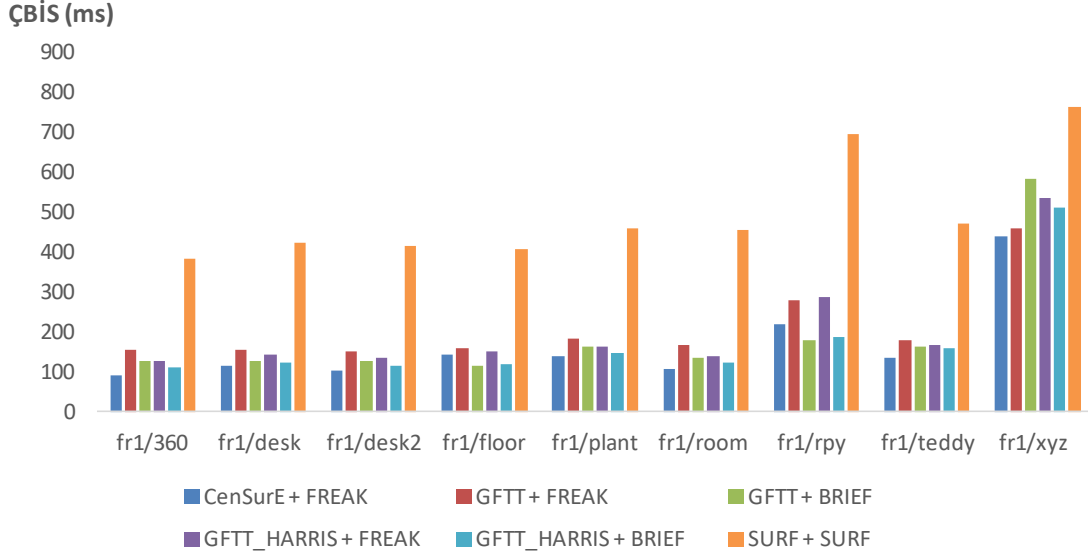
Hız ve doğruluk oranları birlikte ele alındığında en uygun seçenek, ortalama 3.79 cm hata ve 165.7 ms çerçeve başına işlem zamanı ile CenSurE+FREAK kombinasyonudur. Bu kombinasyondan sonra en iyi seçenekler ise GFTT+FREAK ve GFTT+BRIEF metotlarıdır. Diğer yandan, Harris bulucusunun aktif hale getirilmesi GFTT ile elde edilen çalışma hızında artış sağlamasına rağmen hata oranını da bir miktar yükseltmiştir. Aday parametresinin değerinin artması ile işlem süresinde artış gözlenmektedir çünkü hem odometri tahmini hem de döngü kapama tespiti için daha fazla sayıda çerçeve ile dönüşüm hesaplanmaktadır. Diğer yandan, aday sayısının artması genellikle kayma hatasının azalmasını sağlamaktadır (Bkz. Çizelge 4.3). CenSurE+FREAK, SURF+FREAK ve GFTT+BRIEF gibi kombinasyonlar için bazı durumlarda aday sayısı artışıyla

başarıda düşüş gözlenmiştir. Bu durumun temel sebebi, daha çok döngü kapama adayı kullanılması sonucunda bazı durumlarda yanlış döngü kapama tespiti yapılmasıdır.

Çizelge 4.4. *fr1* kayıtları için çerçeve başına işlem süresi (ms)

Nitelik Bulucu	Nitelik Tanımlayıcı	Aday Çerçeve Sayısı			Ortalama
		4	8	12	
BRISK	BRIEF	118.2	177.5	212.0	169.2
BRISK	FREAK	125.3	190.2	236.5	184.0
BRISK	BRISK	125.8	191.7	256.7	191.4
FAST	BRIEF	119.4	180.6	196.0	165.3
FAST	FREAK	119.4	168.7	255.1	181.1
GFTT	BRIEF	135.4	192.6	246.0	191.3
GFTT	FREAK	130.1	213.2	282.5	208.6
GFTT_HARRIS	BRIEF	131.5	176.2	223.3	177.0
GFTT_HARRIS	FREAK	129.6	210.0	275.7	205.1
ORB	BRIEF	91.5	119.3	121.0	110.6
ORB	FREAK	<b>83.6</b>	<b>86.7</b>	<b>89.5</b>	<b>86.6</b>
ORB	ORB	95.5	108.9	131.5	112.0
SIFT	BRIEF	187.5	233.7	260.3	227.2
SIFT	FREAK	186.5	231.0	290.9	236.1
SIFT	SIFT	281.5	334.3	338.5	318.1
CenSurE	BRIEF	124.2	168.0	219.5	170.6
CenSurE	FREAK	120.5	167.1	209.4	165.7
SURF	BRIEF	252.4	294.1	312.6	286.4
SURF	FREAK	236.0	275.2	319.6	276.9
SURF	SURF	434.9	498.6	558.4	497.3
Ortalama		161.4	210.9	251.8	208.0

Şekil 4.4'te ise en başarılı 6 yöntem için kayıt bazında çerçeve başına harcanan ortalama süreler yer almaktadır. *fr1/xyz* ve *fr1/rpy* kayıtları dışında genel olarak birbirine yakın değerler elde edilmiştir. *fr1/xyz* ve *fr1/rpy* kayıtlarının basit ve ufak çaplı algılayıcı hareketlerinden oluşması, işlenen her yeni çerçeve ile arasında geçerli bir dönüşüm hesaplanabilen aday çerçeve sayısını daha hızlı arttırmaktadır. Bu sebeple çizgeye eklenen kenar sayısı daha hızlı artmakta, dolayısı ile her bir çerçeve için işlem süresi yükselmektedir.



Şekil 4.4. En başarılı 6 yöntemle ait, *fr1* kayıtları için çerçeve başına işlem süreleri (Aday parametresinin 4, 8 ve 12 değerleri için elde edilen sonuçların ortalaması)

Tüm nitelik bulucularla birlikte kullanılan BRIEF ve FREAK tanımlayıcıları kıyaslandığında, uygulandıkları 8'er farklı kombinasyonun doğruluk oranlarına göre BRIEF ortalama 5.27 cm, FREAK ise 7.04 cm hata üretmiştir (Bkz. Çizelge 4.3). Hız verilerine bakıldığında ise BRIEF ile çerçeve başına ortalama 187.2 ms süre harcanırken, FREAK için bu değer 193 ms olarak hesaplanmıştır. Kendi tanımlayıcılarına sahip olan SIFT, SURF, BRISK ve ORB nitelik bulucuları BRIEF ya da FREAK tanımlayıcıları ile birlikte kullanıldıklarında genellikle işlem süresi azalmaktadır. İkili tanımlayıcıların oluşturdukları vektörler sadece 0 ve 1 değerlerinden oluştuğu için, kayan noktalı değerlerden oluşan vektörlere göre daha hızlı biçimde karşılaştırma yapılabilmektedir. Dolayısıyla nitelik eşleştirmesi daha hızlı gerçekleştirilebilmektedir. Diğer yandan, SIFT ve SURF bulucuları kendi tanımlayıcıları ile beraber kullanıldığında daha doğru sonuçlar elde edilmiştir.

#### 4.5. Değerlendirme

Bu bölümde, çeşitli nitelik bulucu ve tanımlayıcıların RGB-D algılayıcı kullanan bir SLAM sisteminin performansı üzerindeki etkileri hız ve başarımlar açısından analiz edilmiştir. Gerçekleştirilen deneyler neticesinde, doğruluk ve hız açısından birlikte değerlendirme yapıldığında en uygun seçenek, ortalama olarak çerçeve başına 165.7 ms süre harcayan ve 3.79 cm hata oranına sahip CenSurE+FREAK ikilisidir.



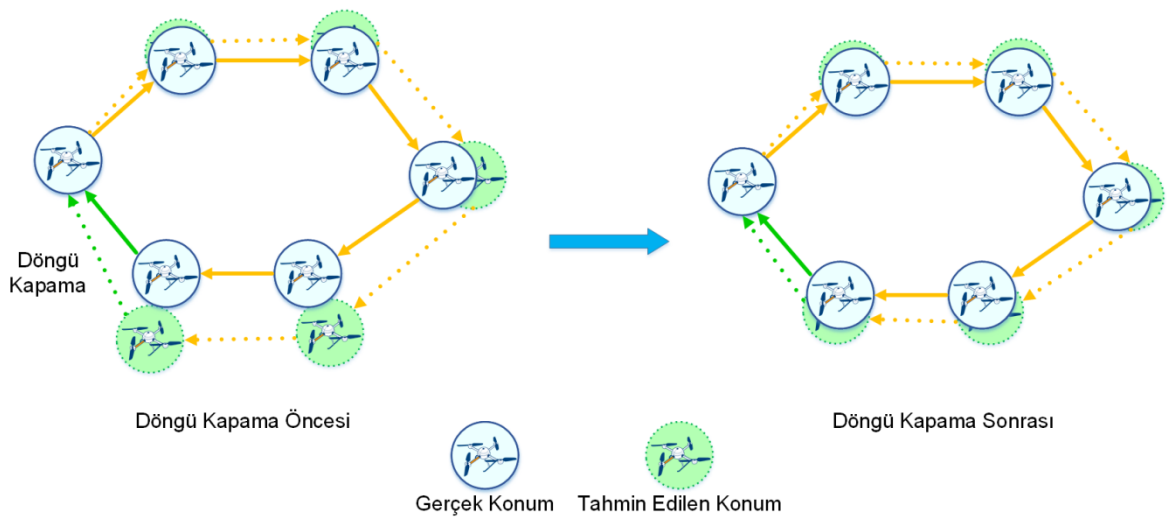
Nitelik bulucu ve nitelik tanımlayıcı tercihinin sonuca doğrudan tesir eden bir faktör olduğu görülmüştür. İşlenen çerçeve üzerinde az sayıda nitelik noktası tespit edilmesi, ortamdaki ışık seviyesinin ani değişimi, çerçevedeki gürültü gibi etkenler başarıyı olumsuz yönde etkilemektedir. Öte yandan algılayıcının hızlı hareket ettirilmesi ya da titretilmesi gibi durumlar da bulanıklığa sebep olarak hata oranını yükseltmektedir.

Bu çalışmada elde edilen sonuçlardan hareketle, ileriki çalışmalarda çerçevelerdeki anahtar noktaların elde edilmesi için CenSurE nitelik bulucusu, noktalara ait tanımlayıcı vektörlerin oluşturulması için ise FREAK nitelik tanımlayıcısı kullanılmıştır.

## 5. GENEL VE YEREL GÖRÜNTÜ NİTELİKLERİ KULLANILARAK DÖNGÜ KAPAMA TESPİTİ

SLAM bağlamında, robotun ortam içerisinde daha önce bulunduğu bir bölgeden tekrar geçmekte olduğunun anlaşılabilmesi yeteneği döngü kapama tespiti olarak adlandırılır.

Döngü kapamaların doğru şekilde tespit edilmesi, SLAM sisteminin başarımı açısından kritik öneme sahiptir. Gezinim esnasında daha önce algılanmış olan bir bölgenin yanlış şekilde yorumlanarak yeni bir bölge olarak yorumlanması, oluşturulan haritanın tutarsız olmasına sebebiyet verebilmektedir. Diğer yandan, döngü kapama tespitinin başarılı şekilde gerçekleştirilebilmesi, odometri tahmininden kaynaklı kayma hatalarının azaltılmasına imkân vermektedir. Özellikle algılayıcıdan elde edilen verilerdeki belirsizlikler odometri hatalarına sebep olmaktadır. Gezinim esnasında robot yolu artırımsal olarak tahmin edildiği için (her bir konum, bir önceki konuma dayalı olarak hesaplandığından), yoldaki kayma miktarı gittikçe artmaktadır. Bu durum da haritanın hatalı şekilde oluşturulmasına neden olmaktadır. Doğru döngü kapama tespiti, odometri hatalarından kaynaklı yol kaymalarının azaltılmasını sağlamaktadır (Bkz. Şekil 5.1). Haritanın genel tutarlılığının sağlanabilmesi açısından, döngü kapama tespiti sonucunda çizgeye eklenen döngü kapama kısıtları optimizasyon için temel oluşturur.



Şekil 5.1. Kayma hatasının döngü kapama tespiti ile azaltılması

Özellikle birbirlerinden uzak düğümler arasında tespit edilen geniş döngü kapamalar kayma hatalarını önemli ölçüde azaltabilmektedir. Diğer taraftan bakıldığında ise, yanlış döngü kapama tespitleri (*false positive*) sistem başarımını önemli derecede düşürebilmektedir. Çünkü birbirleriyle alakası olmayan konumlar arasında oluşturulacak hatalı döngü kapama kısıtları optimizasyon işleminin başarısız olmasına ve yanlış bir çözüme yakınsamasına sebep olabilmektedir.

Döngü kapama tespitinin başarılı şekilde yapılabilmesi için döngü kapama adayları iyi seçilmelidir. RGB-D kamera gibi görsel algılayıcı kullanan SLAM sistemlerinde, döngü kapama adaylarının doğru şekilde seçilebilmesi için etkili bir mekân tanıma yöntemine ihtiyaç vardır. Mekan tanıma problemi; sahne görünümündeki değişimler, algısal örtüşme (*perceptual aliasing*) ve aynı bölge için farklı görüş açıları (*viewpoint*) gibi temel zorluklar sebebiyle çözülmesi zor bir problemdir [7]. Mekân tanımanın doğasında yer alan zorlukların yanında bir diğer büyük problem, robotun ortamdaki gezinimiyle birlikte büyüyen arama uzayıdır. Haritalanan ortam genişledikçe, gittikçe artan yol kaymasından dolayı döngü kapama tespiti daha da önemli hale gelmektedir. Ancak mekân tanımanın doğru yapılabilmesi gittikçe daha da zorlaşmaktadır, çünkü daha fazla gözlem verisi içerisinde arama yapılmaktadır. Dolayısıyla doğru döngü kapama tespiti daha zor hale gelmekte ve daha fazla işlemsel yük getirmektedir. Bu önemli probleminden dolayı, literatürde yer alan RGB-D tabanlı SLAM sistemlerinin büyük çoğunluğu geniş ölçekli ortamlarda çalışmamaktadır.

Bu çalışmada [79], SLAM sistemleri için bir döngü kapama tespiti yöntemi geliştirilmiştir. Nitelik tabanlı SLAM sistemlerinde kolaylıkla uygulanabilen bu yöntem, geniş ölçekli ortamların haritalanabilmesi için RGB-D SLAM sistemine [44] entegre edilmiştir.

### **5.1. Döngü Kapama Tespiti Yönünden SLAM Yöntemlerinin Değerlendirilmesi**

RGB-D algılayıcı tabanlı SLAM sistemleri incelendiğinde, döngü kapama tespiti için uygulanan yöntemler genel olarak; tüm anahtar çerçevelerle karşılaştırma gerçekleştirilmesi [23], önceki çerçevelerden örnekleme yapılması [42], [44], [45], [59], metrik en yakın komşu araması [51] ve kelime çantası yaklaşımı uygulanması [28], [38], [46], [47], [57], [58] şeklinde sıralanabilir. Artırımsal haritalama yaklaşımları [33], [52]–[55], [63] döngü kapama tespiti gerçekleştirmedikleri için

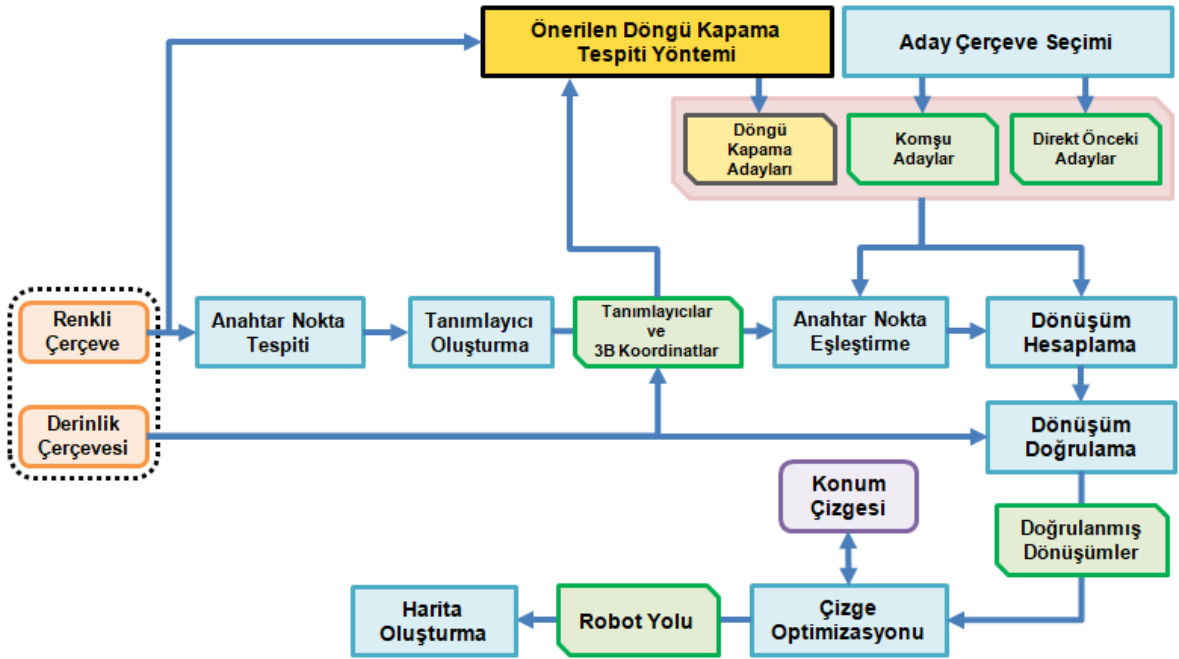
artan kayma hatalarını düzeltmemektedirler. Dolayısıyla artırimsal yaklaşımlar geniş ölçekli haritalama için uygun değildirler. Tüm anahtar çerçevelerle çakıştırma yapılması işlem maliyetini çok fazla arttırmaktadır çünkü anahtar çerçeve sayısı hızlı şekilde ve sürekli artmaktadır. Döngü kapama adaylarının belirli bir uzaklıktaki komşular arasında aranması işlemsel yükü azaltabilmektedir ancak geniş ölçekli döngü kapamaların tespit edilmesi için uygun olmayan bir yaklaşımdır. Çerçevelerden örnekleme yapılması küçük ortamlar için başarılı olabilmektedir çünkü çerçeve sayısı düşüktür. Ancak orta genişledikçe doğru adayların seçilme ihtimali azalmaktadır. Dolayısıyla, örnekleme yapılması geniş ortam haritalamasında döngü kapama tespiti için elverişli bir yöntem değildir. Kelime çantası yaklaşımı [31], [80], bir görsel sözlük (*visual vocabulary*) yapısının oluşturulmasını gerektirir. Bu sözlük yapısı, görsel görüntü niteliklerinden elde edilen görsel kelimelerden oluşur. Nitelik uzayı bu şekilde görsel kelimelere ayrılarak görüntüler bu kelimelerle ifade edilir. Görsel sözlük, çevrimdışı ön-eğitim (*pre-training*) yoluyla [48], [81] ya da çevrimiçi artırimsal şekilde [82], [83] oluşturulabilir. Ön-eğitim yoluyla oluşturulan sözlük statik olduğundan, çevrimdışı yöntemler eğitimin gerçekleştirildiği ortama bağımlıdır ve farklı ortamlarda artan hatalı sonuçlara sebep olmaktadır. Çevrimiçi yöntemler ise sözlüğü haritalanan ortamdaki gezinim süresince artırimsal şekilde oluşturmakta ve güncellemektedirler. Ancak bu işlem maliyetlidir ve geniş ölçekli haritalama için çok uygun değildir.

## 5.2. Döngü Kapama Tespiti Açısından RGB-D SLAM Sistemi

RGB-D SLAM [44] küçük ortamlarda başarılı şekilde haritalama yapabilmektedir. Ancak sistem geniş ortamların haritalanabilmesi için uygun değildir çünkü döngü kapama tespiti rastgele şekilde gerçekleştirilmektedir.

Sistem, görsel bir algılayıcı kullanmasına rağmen döngü kapama adaylarını belirlerken herhangi bir görsel kriter uygulamamakta, dolayısıyla sahne benzerliğini hesaba katmamaktadır. Döngü kapama adayları rastgele seçildiği için, sistem başarımını belirleyen temel faktör mevcut çerçeveye benzer eski çerçevelerin seçilme olasılığıdır. Bu yaklaşım, küçük çaplı ortamların kısa süreli robot gezinimi ile haritalanmasında başarılı sonuçlar üretmektedir. Çünkü anahtar çerçeve sayısı az ve dolayısıyla mevcut çerçeveye benzer adayların seçilme olasılığı nispeten yüksektir. Diğer taraftan, haritalanan ortam nadir döngü

kapamalarla genişledikçe, anahtar çerçeve sayısı sürekli artmakta ve doğru döngü kapama adaylarının seçilme olasılığı önemli ölçüde azalmaktadır. Bu durum, ortam büyüdükçe tutarsız haritalama sonuçlarına sebep olmaktadır. Bu problemin üstesinden gelebilmek için, geliştirilen yöntem RGB-D SLAM sistemine [44] entegre edilerek döngü kapama adaylarının bu yöntem ile belirlenmesi sağlanmıştır (Bkz. Şekil 5.2).



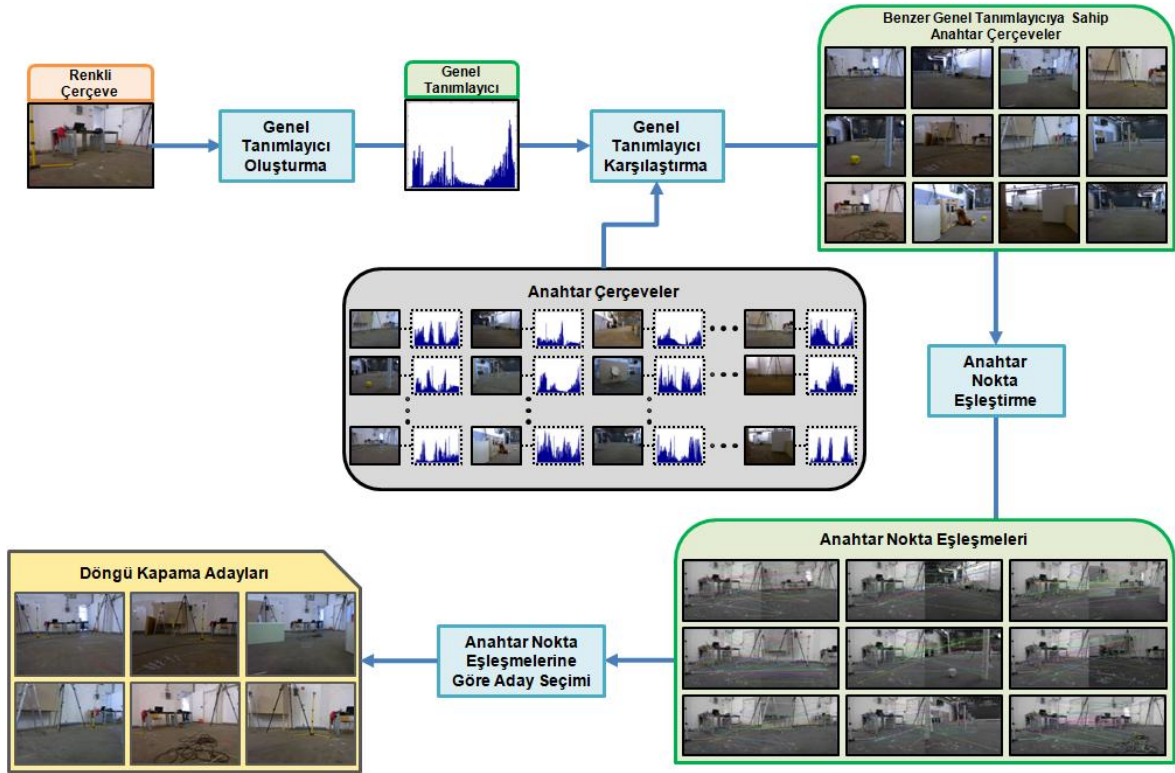
Şekil 5.2. Önerilen döngü kapama tespiti yönteminin RGB-D SLAM sistemi [44] ile entegrasyonu

### 5.3. Önerilen Döngü Kapama Tespiti Yöntemi

Geliştirilen yöntemde, genel ve yerel görüntü niteliklerinden faydalanan bir görsel mekân tanıma yaklaşımı kullanılarak döngü kapama adayları belirlenmektedir. Yöntemin nitelik tabanlı SLAM sistemlerinde gerçekleştirimi oldukça kolaydır. Yöntem iki-geçişli bir algoritma uygulamaktadır. İlk olarak, düşük bellek ve işlem maliyetiyle arama uzayının filtrelenmesi için genel görüntü tanımlayıcıları karşılaştırılmakta ve görsel olarak mevcut çerçeveye benzer bir grup aday (başlangıç grubu) belirlenmektedir. Sonrasında, bu grup üzerinde mevcut çerçeve ile anahtar nokta eşleştirme yapılarak, yerel niteliklerin kullanımıyla döngü kapama adayları belirlenmektedir. Anahtar nokta eşleştirme işleminde, odometri tahmini için hali hazırda elde edilmiş olan anahtar noktalar kullanılmaktadır.

Yöntem anahtar çerçeve mekanizmasını kullanmaktadır, çünkü her yeni çerçeve için önceki çerçevelerin tamamının işlenmesi çok fazla işlem maliyeti getirmektedir. Döngü kapama adaylarının daha önce işlenmiş olan tüm çerçeveler içerisinde aranması yerine, tüm robot yolunu kapsayan bir çerçeve alt kümesinde (anahtar çerçeveler) arama yapılmaktadır.

Yöntemde, birbirini tamamlayan parçalardan oluşan iki farklı teknik tanımlanmıştır. İlk teknikte sadece genel tanımlayıcı benzerliği kullanılmakta olup, daha hızlı döngü kapama tespiti hedeflenmiştir. İkinci teknikte ise anahtar nokta eşleştirmesi de uygulanmaktadır ve daha maliyetli ancak daha başarılı sonuçlar amaçlanmıştır. Geliştirilen yöntemin genel görünümü Şekil 5.3'teki gibidir.



Şekil 5.3. Önerilen döngü kapama tespiti yönteminin genel işleyişi

### 5.3.1. Genel Tanımlayıcı Kullanılarak Mekân Tanıma

Bu teknikte döngü kapama adayları yalnızca genel tanımlayıcı kullanılarak seçilmektedir. Genel tanımlayıcılar görüntünün tamamını işler ve görüntüyü bir bütün olarak sıkı (*compact*) bir yapıyla tanımlarlar. Geliştirilen yöntemde genel tanımlayıcı olarak *histogram* ve *autocorrelogram* kullanılmaktadır. Arama uzayı filtrelenerek benzer bir aday grubu bulunması için, öncelikli olarak bu genel tanımlayıcılar ile ayrı ayrı karşılaştırma yapılmaktadır.

### 5.3.1.1. Histogram Karşılaştırması

Görüntü histogramları [84], görüntüler arasındaki benzerliğin ölçülmesi için yaygın olarak kullanılmaktadırlar. Bu veri yapısı, görüntüdeki piksellerin yoğunluk değerlerinin (*intensity*) genel sıklık dağılımını içerir. Diğer bir deyişle histogram, görüntüde her bir yoğunluk seviyesine ait kaç adet piksel yer aldığı bilgisini gösterir. Yoğunluk değerleri, renkli histogram için her bir renk kanalındaki seviyeleri, gri tonlamalı histogram için ise gri tonlarını ifade ederler. Görüntü histogramları kolayca hesaplanabilirler ve küçük çaplı görüş açısı değişimlerine karşı dayanıklıdırlar.

Görüntü histogramları *bin* değerlerinden oluşurlar ve her bin belirli bir yoğunluk aralığını ifade eder. Görüntüdeki her bir pikselin yoğunluk değerine bakılır ve o yoğunluk seviyesinin dâhil olduğu bin ile ilişkilendirme yapılır. Dolayısıyla tüm pikseller işlendikten sonra, her bir bin kendi yoğunluk aralığına düşen piksel sayısını içerir.

Önerilen yöntemde, histogram benzerliğinin ölçülmesi için 6 farklı uzaklık metriği kullanılmıştır.  $I$  ve  $I'$  karşılaştırılacak görüntüler,  $H$  ve  $H'$  ise sırasıyla bu görüntülerin histogramları olmak üzere,  $N$  histogramlardaki bin sayısını ifade etsin. Dolayısıyla  $H_i$  ve  $H'_i$  histogramların  $i$ . binlerini gösterebilir. Ayrıca  $\bar{H}$  ve  $\bar{H}'$  histogramların ortalamalarını içersin ve  $\{1,2,3, \dots, N\}$  kümesi  $[N]$  ile ifade edilsin.

Buna göre kullanılan metrikler aşağıdaki gibidir;

- Chi-Square uzaklığı

$$d_{h_{Chi-Square}}(I, I') = \sum_{i \in [N]} \frac{(H_i - H'_i)^2}{H_i} \quad (5.1)$$

Bir histogramın dağılımının diğeriyle ifade edilmesine dair uyumsuzluğu ölçer.

- Correlation uzaklığı

$$d_{h_{Correlation}}(I, I') = \frac{\sum_{i \in [N]} (H_i - \bar{H})(H'_i - \bar{H}')}{\sqrt{\sum_{i \in [N]} (H_i - \bar{H})^2 \sum_{i \in [N]} (H'_i - \bar{H}')^2}} \quad (5.2)$$

Benzerlik ölçümü için normalleştirilmiş çapraz ilişkiyi (*normalized cross correlation*) temel alır.

- Euclidean uzaklığı

$$d_{h_{Euclidean}}(I, I') = \sqrt{\sum_{i \in [N]} (H_i - H'_i)^2} \quad (5.3)$$

Etkilliliği nedeniyle yaygın olarak kullanılan bir başka metriktir.

- Hellinger uzaklığı

$$d_{h_{Hellinger}}(I, I') = \sqrt{1 - \frac{1}{\sqrt{H H' N^2}} \sum_{i \in [N]} \sqrt{H_i H'_i}} \quad (5.4)$$

Histogramlardaki daha küçük bin değerlerine olan duyarlılığı arttırmaktadır [85].

- Intersection uzaklığı

$$d_{h_{Intersection}}(I, I') = \sum_{i \in [N]} \min(H_i, H'_i) \quad (5.5)$$

Aynı yoğunluk aralığında yer alan piksel sayısını hesaplar. Kısmi örtüşmeleri ele alabilmektedir [84].

- Manhattan uzaklığı

$$d_{h_{Manhattan}}(I, I') = \sum_{i \in [N]} |H_i - H'_i| \quad (5.6)$$

Basitliği ve modele aykırı değerlere karşı dayanıklılığıyla histogram karşılaştırması için yaygın olarak kullanılmaktadır.

### 5.3.1.2. Autocorrelogram Karşılaştırması

Görüntü *correlogramı*, Huang ve diğerleri [86] tarafından görüntü erişim (*image retrieval*) sistemleri için önerilmiş bir genel niteliklerdir. Correlogram, temel olarak görüntüdeki renkler arasındaki uzamsal ilişkilere odaklanır ve bu ilişkileri genel olarak tanımlar. Correlogramların hesaplanmaları kolaydır ve görüş açısı değişimlerine karşı histogramlardan daha dayanıklıdır.

Görüntü correlogramı, iki renk değerinin arasında belirli bir uzaklık bulunan piksel konumlarında yer alma olasılıklarını içeren bir tablodan oluşur.  $I$  görüntüsü  $r \times r$  boyutunda olsun ve nicelendirme (*quantization*) sonucunda  $m$  adet renk  $(c_1, \dots, c_m)$  içersin.  $I(p)$  ise,  $(x, y) \in I$  koordinatlarında yer alan  $p$  pikselinin içerdiği renk olan  $c$ 'yi ifade etsin. Dolayısıyla  $I_c = \{p | I(p) = c\}$  ise,  $p \in I_c$  ifadesi  $p \in I$  ve  $I(p) = c$  anlamına gelir.  $p_1 = (x_1, y_1)$  ve  $p_2 = (x_2, y_2)$  pikselleri arasındaki uzaklık  $|p_1 - p_2| = \max\{|x_1 - x_2|, |y_1 - y_2|\}$  şeklinde hesaplanır.  $[r]$  ifadesi  $\{1, 2, 3, \dots, r\}$  kümesini ifade eder.



Uzaklık değeri  $d \in [r]$  önceden belirlenmiş (*fixed a priori*) olsun.  $I$  görüntüsünün correlogramı aşağıdaki gibi tanımlanır;

$$\gamma_{c_i, c_j}^{(t)}(I) = \Pr_{p_1 \in I_{c_i}, p_2 \in I} [p_2 \in I_{c_j} \mid |p_1 - p_2| = t] \quad (5.7)$$

Burada  $i, j \in [m]$  ve  $t \in [d]$  geçerlidir. Dolayısıyla  $\gamma_{c_i, c_j}^{(t)}$  ifadesi,  $c_i$  rengine sahip herhangi bir pikselden  $t$  uzaklıktaki bir pikselde  $c_j$  renginin bulunma olasılığını ifade eder.

Correlogramın özelleşmiş bir versiyonu, sadece aynı renk değerlerine sahip pikseller arasındaki uzamsal ilişkiyi ele alan *autocorrelogramdır*.  $I$  görüntüsünün autocorrelogramı aşağıdaki gibi tanımlanır;

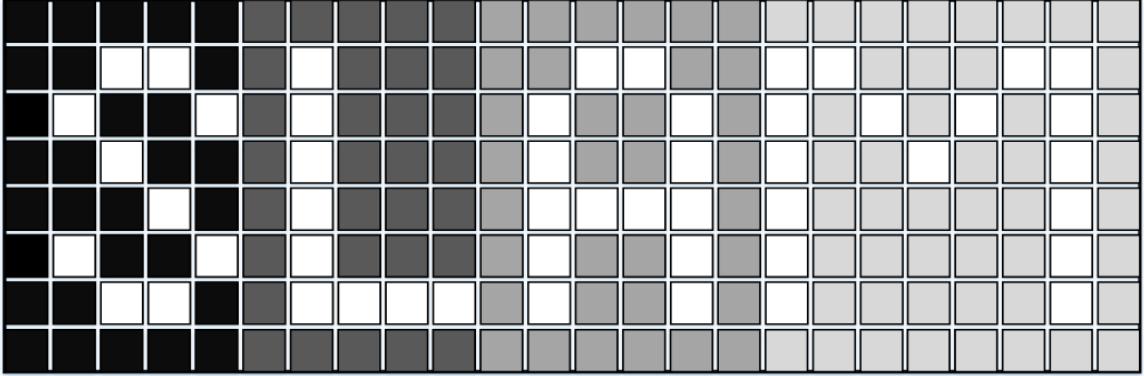
$$\alpha_c^{(t)}(I) = \gamma_{c, c}^{(t)}(I) \quad (5.8)$$

Dolayısıyla autocorrelogram, görüntü içerisinde birbirinden belirli bir uzaklıkta bulunan herhangi iki pikselin aynı renk değerine sahip olma olasılıklarını içerir.

Geliştirilen yöntemde autocorrelogram yapısı tercih edilmiştir çünkü correlogram  $O(m^2d)$  bellek karmaşıklığına sahipken, autocorrelogram  $O(md)$  kadar alan gerektirmektedir. Şekil 5.4'te,  $8 \times 24$  boyutunda ve 8 gri tonu seviyesinden oluşan gri tonlamalı bir görüntüye ait autocorrelogramın hesaplanma aşamaları görülmektedir.

Gri Tonu Seviyeleri

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---



Gri Tonlamalı Görüntü

0	0	0	0	0	2	2	2	2	2	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6
0	0	7	7	0	2	7	2	2	2	5	5	7	7	5	5	7	7	6	6	6	7	7	6	6
0	7	0	0	7	2	7	2	2	2	5	7	5	5	7	5	7	6	7	6	7	6	7	6	6
0	0	7	0	0	2	7	2	2	2	5	7	5	5	7	5	7	6	6	7	6	6	7	6	6
0	0	0	7	0	2	7	2	2	2	5	7	7	7	7	5	7	6	6	6	6	6	7	6	6
0	7	0	0	7	2	7	2	2	2	5	7	5	5	7	5	7	6	6	6	6	6	7	6	6
0	0	7	7	0	2	7	7	7	7	5	7	5	5	7	5	7	6	6	6	6	6	7	6	6
0	0	0	0	0	2	2	2	2	2	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6

Gri Tonu Değerleri

7	0	0
0	7	7
0	0	0

2/8

7	2	2
7	7	7
2	2	2

3/8

5	7	6
5	7	6
5	6	6

1/8

6	7	6
6	7	6
6	6	6

1/8

$$\frac{\dots + 2/8 + \dots + 3/8 + \dots + 1/8 + \dots + 1/8}{50}$$

0	1	2	3	4	5	6	7
0.645	0	0.576	0	0	0.515	0.708	0.270

Autocorrelogram

Şekil 5.4. Autocorrelogram hesaplama örneği

Autocorrelogram benzerliğinin hesaplanması için, Huang ve diğerleri [86] tarafından da önerildiği üzere 2 farklı uzaklık metriği kullanılmıştır.  $I$  ve  $I'$  karşılaştırılacak görüntüler olmak üzere, kullanılan metrikler şu şekildedir;

- Manhattan uzaklığı

$$d_{\alpha_{Manhattan}}(I, I') = \sum_{i \in [m], t \in [d]} |\alpha_{c_i}^{(t)}(I) - \alpha_{c_i}^{(t)}(I')| \quad (5.9)$$

- Relative uzaklığı

$$d_{\alpha_{Relative}}(I, I') = \sum_{i \in [m], t \in [d]} \frac{|\alpha_{c_i}^{(t)}(I) - \alpha_{c_i}^{(t)}(I')|}{1 + \alpha_{c_i}^{(t)}(I) + \alpha_{c_i}^{(t)}(I')} \quad (5.10)$$

Bu metrik, bileşenler arasındaki göreceli mesafeyi ağırlıklandırma yoluyla hesaplamaktadır.

### 5.3.1.3. Döngü Kapama Adaylarının Belirlenmesi

Önerilen yöntemde kullanılacak genel tanımlayıcı, tercihe göre 3 kanallı RGB histogram, tek kanallı gri tonlamalı histogram, RGB autocorrelogram ya da gri tonlamalı autocorrelogram olabilir. Autocorrelogram hesaplanmasında, görüntü öncelikle belirli bir renk / gri seviyesi sayısına göre nicelendirilir.

---

**Input** : mevcut çerçeve  $C$   
önceki anahtar çerçeveleri içeren küme  $K$

**Output**: döngü kapama adayları

**Parameters**: *uzaklık\_metriği*, *aday\_sayısı*

$H_c \leftarrow$  mevcut çerçeve  $C$  için genel tanımlayıcıyı oluştur

**for** her bir anahtar çerçeve  $F_k \in K$  **do**

$S_k \leftarrow$  anahtar çerçeveye ait genel tanımlayıcı  $H_{F_k}$  ile  $H_c$  arasındaki *uzaklık\_metriği* türünden uzaklık

$L \leftarrow$  anahtar çerçeve  $F_k$ 'yi benzerlik skoru  $S_k$ 'ya göre sıralı olarak listeye ekle

**end**

**return**  $L$ 'deki ilk *aday\_sayısı* adet anahtar çerçeve

---

Şekil 5.5. Genel tanımlayıcı kullanılarak aday seçimi

Algılayıcıdan gelen her yeni çerçeve için görüntüyü ifade eden genel tanımlayıcı oluşturulur. Oluşturulan genel tanımlayıcı, geçmiş çerçeveler için her seferinde tekrar hesaplanmaması için bellekte saklanır. Sonrasında ise, oluşturulan genel tanımlayıcı ile tüm anahtar çerçevelerin genel tanımlayıcıları arasında yukarıda tanımlanan uzaklık metriklerinden birine göre karşılaştırma yapılır. Karşılaştırma sonucunda anahtar çerçeveler, mevcut çerçeveye benzerliklerine göre sıralanır. Mevcut çerçeveye en benzer *aday\_sayısı* kadar anahtar çerçeve döngü kapama

adayları olarak seçilir (Bkz. Şekil 5.5). Bu işlem çok düşük maliyetlidir çünkü genel tanımlayıcıların oluşturulması ve karşılaştırılması çok hızlı şekilde gerçekleştirilebilen işlemlerdir.

### 5.3.2. Anahtar Nokta Eşleştirmesi ile Mekân Tanıma

Bu teknikte, genel görüntü niteliklerinin yanında yerel niteliklerin de kullanımıyla döngü kapama adayları belirlenmektedir. Bu bağlamda yerel niteliklerden, görüntüden elde edilen anahtar noktaların çerçeve karşılaştırması aşamasında kullanımı yoluyla faydalanılmıştır. Bölüm 4'te de bahsedildiği üzere yerel nitelikler, görüntüdeki dikkat çekici bölgelerde yer alan belirgin anahtar noktalar etrafındaki alanlara dair bilgileri kapsarlar. Dolayısıyla, bu noktalar için oluşturulan nitelik tanımlayıcı vektörler sahneye dair uzamsal bilgileri içerirler. Nitelik tanımlayıcılar geniş görüş açısı değişimlerine karşı dayanıklıdır. Buradan hareketle, genel nitelikler ile birlikte yerel niteliklerin de kullanımıyla döngü kapama tespiti başarımının artırılması hedeflenmiştir.

Yöntemde, nitelik tanımlayıcıları kullanılarak sahne tanımlanmakta ve *brute-force* anahtar nokta eşleştirmesi yoluyla mekân tanıma gerçekleştirilmektedir. *brute-force* eşleştirme işleminde, karşılaştırılan çerçevelerdeki tüm anahtar nokta çiftleri arasındaki uzaklıklar hesaplanmaktadır. Anahtar noktalar eşleştirilirken Lowe [24] tarafından önerilen uzaklık oranı yaklaşımı kullanılarak eşleştirme başarımının artırılması hedeflenmiştir.

$$\frac{d_{en\_yakın\_anahtar\_nokta}}{d_{en\_yakın\_ikinci\_anahtar\_nokta}} > oran\_eşiği \quad (5.11)$$

Uzaklık oranı yaklaşımında; kaynak görüntüdeki bir anahtar noktanın hedef görüntüdeki en yakın nokta ile eşleştirilebilmesi için, hedef görüntüdeki en yakın ve ikinci en yakın noktalara olan mesafeler arasındaki oranın belirli bir eşik değerden büyük olması gerekmektedir. Bu yaklaşımın temel mantığı; kaynak görüntüdeki bir noktanın hedef görüntüde bulunan yalnızca bir noktaya karşılık gelebileceğinden hareketle, doğru bir eşleşme için hedef görüntüdeki en yakın noktanın ikinci en yakın noktadan çok daha düşük uzaklıkta bulunması gerekliliğidir. Dolayısıyla, yüksek boyutlu nitelik uzayında yer almakta olan en yakın ve ikinci en yakın noktalar için benzer uzaklıklar elde edilmesi, eşleşmede belirsizlik olduğu şeklinde yorumlanmaktadır. Bu yaklaşım, özellikle ortamdaki tekrarlı yapılardan kaynaklı yanlış nokta eşleşmelerinin elenmesi için faydalıdır.

---

```

Input : mevcut çerçeve  $C$ 
          genel tanımlayıcı karşılaştırması ile elde edilen başlangıç anahtar çerçeve grubu  $G$ 
Output: döngü kapama adayları
Parameters:  $oran\_eşiği$ ,  $aday\_sayısı$ 
 $K \leftarrow$  mevcut çerçeve  $C$ 'ye ait nitelik tanımlayıcıları
for her bir anahtar çerçeve  $G_f \in G$  do
   $D_{G_f} \leftarrow$  anahtar çerçeve  $G_f$ 'ye ait nitelik tanımlayıcıları
  for her bir nitelik tanımlayıcısı  $k \in K$  do
    for her bir nitelik tanımlayıcısı  $d \in D_{G_f}$  do
      |  $H_{k-d} \leftarrow k$  ve  $d$  arasındaki Hamming uzaklığını hesapla
    end
     $c \leftarrow G_f$ 'deki en yakın anahtar noktaya olan uzaklık
     $sc \leftarrow G_f$ 'deki ikinci en yakın anahtar noktaya olan uzaklık
    if  $c/sc > oran\_eşiği$  then
      |  $M_{G_f} \leftarrow$  eşleşmeyi  $G_f$ 'nin kabul edilen eşleşmeler kümesine ekle
    end
  end
   $L \leftarrow$  anahtar çerçeve  $G_f$ 'yi,  $M_{G_f}$ 'nin büyüklüğüne göre azalan sırada listeye ekle
end
return  $L$ 'deki ilk  $aday\_sayısı$  adet anahtar çerçeve

```

---

### Şekil 5.6. Anahtar nokta eşleştirmesi ile aday seçimi

Anahtar nokta eşleştirme işlemi, genel tanımlayıcı karşılaştırmasına kıyasla maliyetli bir işlemdir. Bu sebeple anahtar nokta eşleştirmesinin tüm anahtar çerçeveler için uygulanması yerine, genel tanımlayıcı benzerliğine göre filtreleme yapılarak  $gruplama\_faktörü \times aday\_sayısı$  kadar anahtar çerçeve başlangıç grubu olarak seçilmektedir. Sonrasında ise mevcut çerçeve ile bu grupta yer alan tüm anahtar çerçeveler arasında anahtar nokta eşleştirmesi gerçekleştirilmektedir. Anahtar çerçeveler mevcut çerçeve ile aralarında elde edilen nokta eşleşmelerinin sayısına göre sıralanmakta ve en çok eşleşme üreten  $aday\_sayısı$  adet çerçeve döngü kapama adayı olarak seçilmektedir (Bkz. Şekil 5.6).

Nitelik tabanlı bir SLAM sisteminde, anahtar noktaların belirlenmesi ve tanımlayıcıların oluşturulması işlemleri, algılayıcıdan gelen her yeni çerçeve için odometri tahmini aşamasında zaten gerçekleştirilmektedir. Dolayısıyla, bu teknikle döngü kapama adaylarının belirlenmesi için hali hazırda mevcut olan tanımlayıcılar kullanılmaktadır. Bu sebeple, tekniği uygulamanın maliyeti sadece nokta eşleştirme işlemine ait maliyettir.

Önerilen yöntem RGB-D SLAM sistemine [44] entegre edilmiş ve döngü kapama adaylarının anahtar çerçeveler içerisinden rastgele belirlenmesi yerine bu yöntemle seçilmesi sağlanmıştır. İlerleyen kesimde, yöntem içerisinde tanımlanan teknikler yoluyla elde edilen haritalama sonuçları sunulmuştur.

#### 5.4. Deneysel Kurulum

Deneylerde TUM RGB-D Benchmark veri kümesinde yer alan *fr2* kayıtları kullanılmıştır. *fr2* grubunda yer alan kayıtlar içerdikleri uzun robot yolları, geniş döngü kapamalar ve zorlu şartlar ile geniş ortamlarda haritalama performansının ölçülmesi için uygundur. Çizelge 5.1’de, kullanılan *fr2* kayıtlarına dair ayrıntılı bilgiler yer almaktadır.

Çizelge 5.1. Deneylerde kullanılan *fr2* görüntü kayıtları hakkında ayrıntılı bilgiler

Kayıt	Süre (s)	Yol Uzunluğu (m)	Ort. Öteleme Hızı (m/s)
fr2/360_hemisphere	91.5	14.77	0.163
fr2/coke	84.6	11.68	0.140
fr2/desk	99.4	18.88	0.193
fr2/dishes	100.6	15.01	0.151
fr2/flowerbouquet	99.4	10.76	0.109
fr2/flowerbouquet_br.	76.9	11.92	0.157
fr2/large_no_loop	112.4	26.09	0.243
fr2/large_with_loop	173.2	39.11	0.231
fr2/metallic_sphere	75.6	11.04	0.148
fr2/metallic_sphere2	62.3	11.81	0.193
fr2/pioneer_360	72.8	16.12	0.225
fr2/pioneer_slam	155.7	40.38	0.261
fr2/pioneer_slam2	115.6	21.74	0.190
fr2/pioneer_slam3	111.9	18.14	0.164

Deneylerde döngü kapama tespiti ve odometri tahmini için, önceki bölümde elde edilen sonuçlara göre (Bkz. Bölüm 4) hız ve doğruluk açısından en başarılı performansı sergileyen CenSurE nitelik bulucusu ve FREAK nitelik tanımlayıcısı birlikte kullanılmıştır.

Deneylerde kullanılan çeşitli parametreler için uygulanan değerler Çizelge 5.2’de özetlenmiştir. Çerçeveler arasında dönüşüm hesaplanması için RANSAC döngüsü en fazla 250 yineleme gerçekleştirmektedir. Çerçeve başına elde edilen anahtar nokta sayısı 700 ile sınırlandırılmıştır. Nitelik eşleştirme işleminde ise *oran\_eşiği* 0.8 olarak uygulanmakta ve FREAK bir ikili tanımlayıcı olduğu için Hamming mesafesi kullanılmaktadır.

Döngü kapama tespiti için mekân tanıma işleminde, histogramlardaki her bir kanal için bin sayısı 32 olarak seçilmiştir. Autocorrelogram hesaplanması aşamasında ise hem nicelendirme faktörü hem de piksel mesafesi parametreleri için ikişer farklı

değer denenmiştir. Farklı deneyler kapsamında; görüntüler 64 ya da 128 renk/gri tonu şeklinde nicelendirilmiş, autocorrelogramlar ise 1 ya da 2 piksel uzaklığına göre oluşturulmuştur.

Aday çerçeve parametreleri ise  $n = 8$  direkt önceki çerçeve,  $k = 20$  komşu çerçeve ve  $l = 8$  döngü kapama çerçevesi şeklinde seçilmiştir. *gruplama\_faktörü* ise 4 olarak uygulanmıştır. Dolayısıyla, geliştirilen döngü kapama tespiti yöntemi ilk aşamada genel tanımlayıcı karşılaştırmasına göre  $4 \times 8 = 32$  en benzer anahtar çerçeveyi başlangıç grubu olarak belirlemekte, sonrasında ise bu gruptan mevcut çerçeveyle en yüksek sayıda anahtar nokta eşleşmesi üreten *aday\_sayısı* = 8 anahtar çerçeveyi döngü kapama adayı olarak seçmektedir.

Çizelge 5.2. Deneylerde uygulanan parametre değerleri

Parametre	Değer
<b><i>Dönüşüm Parametreleri</i></b>	
Maksimum RANSAC yinleme sayısı	250
Maksimum anahtar nokta sayısı	700
<i>oran_eşiği</i>	0.8
<b><i>Mekan Tanıma Parametreleri</i></b>	
Histogram bin sayısı	32
Nicelendirme faktörü	64,128
Piksel mesafesi	1,2
<i>gruplama_faktörü</i>	4
<b><i>Aday Parametreleri</i></b>	
Direkt önceki aday sayısı ( $n$ )	8
Komşu aday sayısı ( $k$ )	20
Döngü kapama adayı sayısı ( $l$ )	8

Uygulanan parametre değerleri, gerçekleştirilen çeşitli deneyler sonucunda elde edilen verilerin analiz edilmesi ile belirlenmiştir. Deneylerin gerçekleştirildiği platform; önceki bölümde de kullanılmış olan, Ubuntu 12.04 işletim sistemine sahip, Intel Core i7-2600 CPU 3.40GHz işlemci ve 8GB RAM bulunduran bir bilgisayardır. Her bir deney 5 defa gerçekleştirilmiş ve elde edilen 5 sonucun ortalaması nihai sonuç olarak sunulmuştur.

## 5.5. Sonular

Bu kesimde dng kapama adaylarının; histogram benzerliđi, autocorrelogram benzerliđi, histogram benzerliđi ile nitelik eŐleŐtirmesi ve autocorrelogram benzerliđi ile nitelik eŐleŐtirmesi teknikleri kullanılarak belirlenmesi neticesinde elde edilen sonular incelenmiŐtir.

### 5.5.1. Histogram Benzerliđi Deneyleri

Deneyler kapsamında ilk olarak dng kapama adaylarının sadece histogram benzerliđi kullanılarak belirlenmesi test edilmiŐtir. Mevcut ereveye en benzer histograma sahip 8 anahtar ereve dng kapama adayı olarak seilmiŐtir. Bu teknikte elde edilen kayma hatası (RMS-ATE) sonuları, RGB ve gri tonlamalı histogram kullanımı iin sırasıyla izelge 5.3 ve izelge 5.4'te sunulmuŐ ve RGB-D SLAM sistemi [44] ile karŐılaŐtırılmıŐtır. Her bir kayıt iin 5 deneme sonucunda elde edilen ortalama RMS-ATE sonucu listelenmiŐ olup, izelgelerin son satırlarında ise stn ortalamaları verilmiŐtir.

izelge 5.3. RGB histogram benzerliđi tekniđi iin mutlak yol hatası (RMS-ATE) sonuları (m)

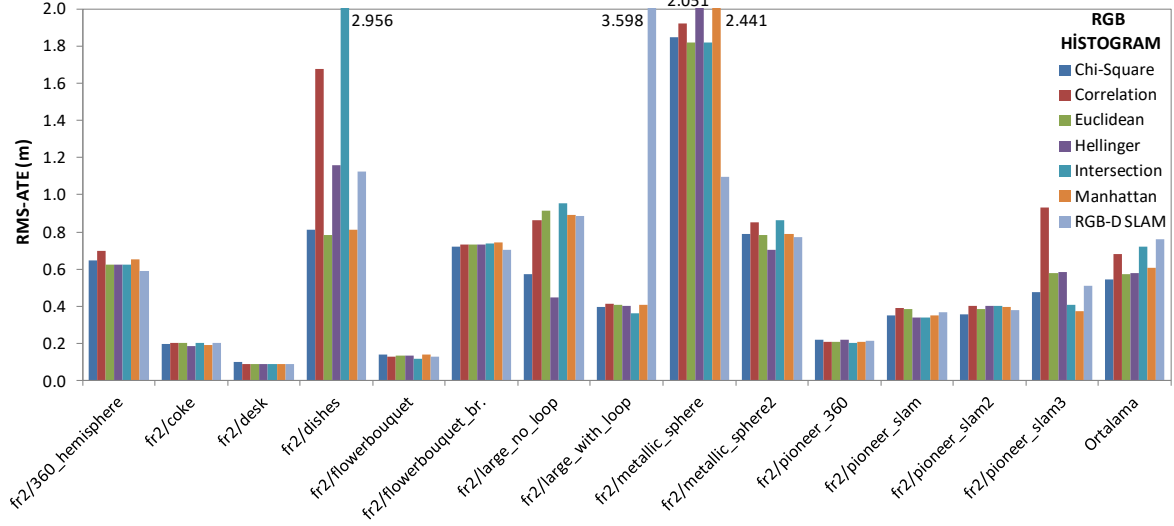
Uzaklık Metriđi	Chi-Square	Correlation	Euclidean	Hellinger	Intersection	Manhattan	RGB-D SLAM
fr2/360_hemisphere	0.649	0.701	0.622	0.626	0.623	0.650	<b>0.592</b>
fr2/coke	0.195	0.201	0.203	<b>0.186</b>	0.204	0.192	0.205
fr2/desk	0.100	0.088	0.089	<b>0.087</b>	<b>0.087</b>	<b>0.087</b>	0.090
fr2/dishes	0.810	1.676	<b>0.784</b>	1.158	2.956	0.809	1.126
fr2/flowerbouquet	0.143	0.129	0.133	0.137	<b>0.120</b>	0.141	0.131
fr2/flowerbouquet_br.	0.720	0.735	0.733	0.732	0.739	0.744	<b>0.703</b>
fr2/large_no_loop	0.573	0.865	0.916	<b>0.450</b>	0.954	0.891	0.887
fr2/large_with_loop	0.394	0.413	0.408	0.400	<b>0.360</b>	0.408	3.598
fr2/metallic_sphere	1.848	1.920	1.817	2.051	1.818	2.441	<b>1.099</b>
fr2/metallic_sphere2	0.789	0.855	0.783	<b>0.702</b>	0.866	0.790	0.772
fr2/pioneer_360	0.219	0.209	0.209	0.218	<b>0.204</b>	0.207	0.213
fr2/pioneer_slam	0.352	0.389	0.387	<b>0.339</b>	0.342	0.351	0.367
fr2/pioneer_slam2	<b>0.357</b>	0.402	0.385	0.404	0.401	0.396	0.381
fr2/pioneer_slam3	0.477	0.931	0.581	0.584	0.407	<b>0.375</b>	0.511
Ortalama	<b>0.545</b>	0.680	0.575	0.577	0.720	0.606	0.762



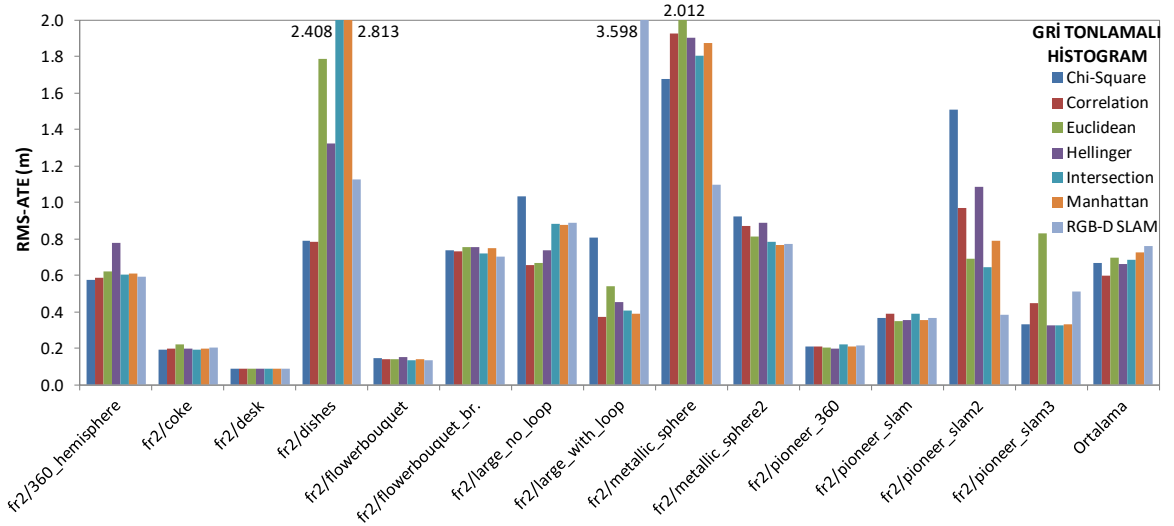
Çizelge 5.4. Gri tonlamalı histogram benzerliği tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m)

Uzaklık Metriği	Chi-Square	Correlation	Euclidean	Hellinger	Intersection	Manhattan	RGB-D SLAM
fr2/360_hemisphere	<b>0.574</b>	0.584	0.622	0.777	0.604	0.610	0.592
fr2/coke	<b>0.189</b>	0.196	0.224	0.198	0.192	0.195	0.205
fr2/desk	0.090	<b>0.089</b>	<b>0.089</b>	<b>0.089</b>	<b>0.089</b>	<b>0.089</b>	0.090
fr2/dishes	0.790	<b>0.784</b>	1.790	1.325	2.408	2.813	1.126
fr2/flowerbouquet	0.148	0.138	0.142	0.151	0.133	0.139	<b>0.131</b>
fr2/flowerbouquet_br.	0.736	0.731	0.753	0.753	0.720	0.751	<b>0.703</b>
fr2/large_no_loop	1.036	<b>0.655</b>	0.670	0.737	0.885	0.877	0.887
fr2/large_with_loop	0.809	<b>0.374</b>	0.540	0.456	0.406	0.388	3.598
fr2/metallic_sphere	1.680	1.926	2.012	1.903	1.805	1.875	<b>1.099</b>
fr2/metallic_sphere2	0.922	0.870	0.815	0.887	0.785	<b>0.768</b>	0.772
fr2/pioneer_360	0.207	0.207	0.204	<b>0.198</b>	0.219	0.212	0.213
fr2/pioneer_slam	0.368	0.391	<b>0.348</b>	0.354	0.390	0.357	0.367
fr2/pioneer_slam2	1.511	0.970	0.690	1.085	0.645	0.788	<b>0.381</b>
fr2/pioneer_slam3	0.332	0.445	0.832	0.324	<b>0.323</b>	0.330	0.511
Ortalama	0.671	<b>0.597</b>	0.695	0.660	0.686	0.728	0.762

Ortalamada en başarılı sonuç, 54.5 cm kayma hatasıyla RGB histogram ve Chi-Square uzaklığının birlikte kullanımıyla (RGB+Chi-Square) elde edilmiştir. RGB+Euclidean ve RGB+Hellinger kombinasyonları ise sırasıyla 57.5 cm ve 57.7 cm hata ile ortalamada en iyi ikinci ve üçüncü performansı sergilemişlerdir. Genel olarak, Correlation ve Intersection metrikleri dışında RGB histogram kullanımı gri tonlamalıya göre daha başarılı sonuçlar üretmiştir. Histogram benzerliği tekniği RGB-D SLAM [44] ile karşılaştırıldığında, döngü kapama adaylarının rastgele seçilmesi yerine bu teknikle belirlenmesi her durumda ortalama hatayı azaltmaktadır. Histogram benzerliği kullanılarak elde edilen en yüksek ortalama hata olan 72.8 cm (Gri+Manhattan), RGB-D SLAM [44]'in ürettiği 76.2 cm'den daha düşüktür.



Şekil 5.7. RGB histogram benzerliği tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması



Şekil 5.8. Gri tonlamalı histogram benzerliği tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması

Şekil 5.7 ve Şekil 5.8'de, RGB ve gri tonlamalı histogram kullanımı ile elde edilen yol hataları grafik üzerinde sunulmuştur. Grafiklerden de gözlendiği üzere, bu teknikteki en başarılı kombinasyon olan RGB+Chi-Square bazı kayıtlar için RGB-D SLAM [44]'den daha hatalı sonuçlar üretmektedir. Genel olarak daha uzun robot yolları içeren kayıtlarda daha iyi sonuç veren bu kombinasyon ile RGB-D SLAM [44] arasındaki ortalama hata farkı büyük oranda *fr2/large\_with\_loop* kaydından

kaynaklanmaktadır. Bu durum, önerilen yöntemin geniş ortamlarda yerel görüntü niteliklerini kullanmadan sadece genel tanımlayıcı tekniğiyle bile etkili olduğunu göstermektedir. Diğer yandan, çoğu kombinasyon ile *fr2/metallic\_sphere* için diğer kayıtlara göre daha yüksek hata oranları gözlenmiştir (Bkz. Şekil 5.7 ve Şekil 5.8). Bu durum, bu kayıt için döngü kapama adaylarının seçiminde sadece renk / gri tonu seviyesi kullanımının yeterli olmadığı anlamına gelmektedir.

### 5.5.2. Autocorrelogram Benzerliği Deneyleri

İkinci grup deneylerde, döngü kapama adayları autocorrelogram benzerliğine göre seçilmiştir. Çizelge 5.5 ve Çizelge 5.6'da, sırasıyla RGB ve gri tonlamalı autocorrelogram için RMS-ATE sonuçları listelenmiş ve RGB-D SLAM [44] ile karşılaştırma yapılmıştır.

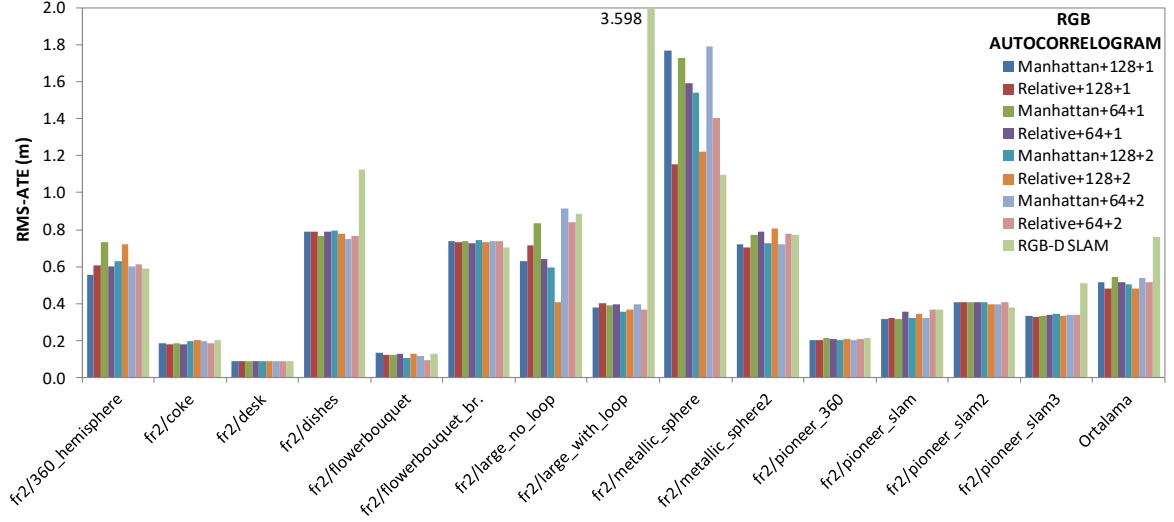
Çizelge 5.5. RGB autocorrelogram benzerliği tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m)

Nicelendirme Faktörü	64				128				RGB-D SLAM
	1		2		1		2		
Piksel Uzaklığı	1		2		1		2		
Uzaklık Metriği	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	
fr2/360_hemisphere	0.734	0.604	0.604	0.615	<b>0.555</b>	0.608	0.631	0.723	0.592
fr2/coke	0.188	0.183	0.198	0.186	0.185	<b>0.180</b>	0.195	0.201	0.205
fr2/desk	<b>0.089</b>	0.090	<b>0.089</b>	<b>0.089</b>	<b>0.089</b>	<b>0.089</b>	0.090	<b>0.089</b>	0.090
fr2/dishes	0.769	0.789	<b>0.747</b>	0.768	0.792	0.790	0.792	0.779	1.126
fr2/flowerbouquet	0.123	0.129	0.117	<b>0.097</b>	0.137	0.125	0.106	0.130	0.131
fr2/flowerbouquet_br.	0.737	0.729	0.735	0.739	0.736	0.734	0.745	0.733	<b>0.703</b>
fr2/large_no_loop	0.835	0.643	0.912	0.841	0.632	0.714	0.598	<b>0.408</b>	0.887
fr2/large_with_loop	0.391	0.396	0.397	0.371	0.381	0.401	<b>0.358</b>	0.370	3.598
fr2/metallic_sphere	1.727	1.593	1.791	1.406	1.769	1.153	1.539	1.220	<b>1.099</b>
fr2/metallic_sphere2	0.770	0.791	0.720	0.779	0.719	<b>0.703</b>	0.726	0.808	0.772
fr2/pioneer_360	0.215	0.209	0.206	0.208	<b>0.201</b>	0.206	0.204	0.209	0.213
fr2/pioneer_slam	<b>0.315</b>	0.355	0.322	0.368	0.318	0.324	0.325	0.346	0.367
fr2/pioneer_slam2	0.407	0.408	0.398	0.410	0.406	0.408	0.408	0.397	<b>0.381</b>
fr2/pioneer_slam3	0.335	0.339	0.343	0.339	0.333	<b>0.331</b>	0.347	0.335	0.511
Ortalama	0.546	0.518	0.541	0.515	0.518	0.483	0.505	<b>0.482</b>	0.762

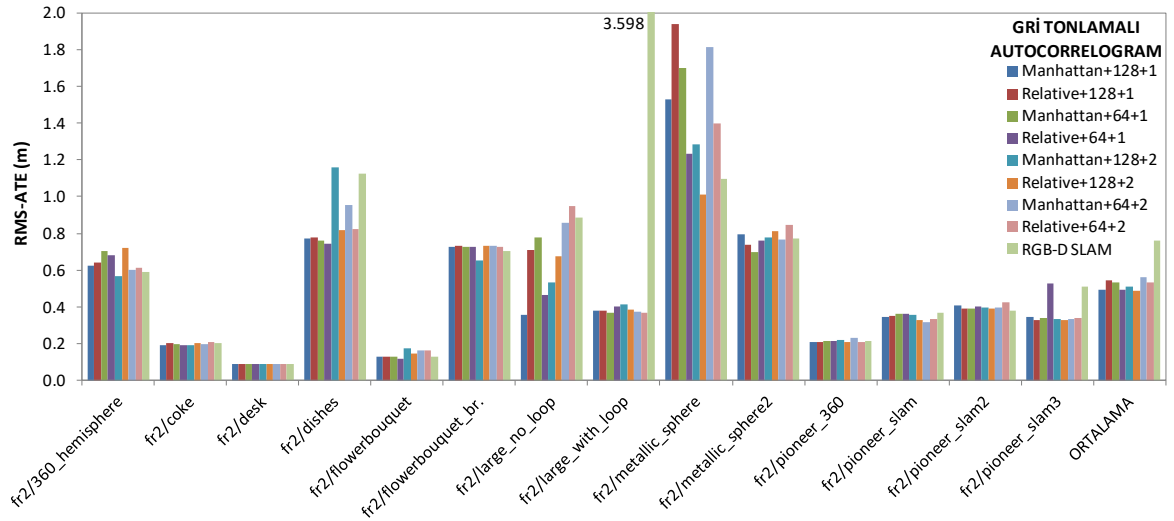
Çizelge 5.6. Gri tonlamalı autocorrelogram benzerliği tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m)

Nicelendirme Faktörü	64				128				RGB-D SLAM
	1		2		1		2		
Piksel Uzaklığı	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	
Uzaklık Metriği	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	
fr2/360_hemisphere	0.706	0.681	0.604	0.612	0.624	0.641	<b>0.565</b>	0.723	0.592
fr2/coke	0.200	0.193	0.199	0.208	<b>0.191</b>	0.203	0.194	0.203	0.205
fr2/desk	0.090	0.090	<b>0.089</b>	<b>0.089</b>	0.090	0.090	<b>0.089</b>	0.090	0.090
fr2/dishes	0.761	<b>0.746</b>	0.953	0.826	0.771	0.780	1.158	0.816	1.126
fr2/flowerbouquet	0.131	<b>0.119</b>	0.163	0.163	0.127	0.131	0.176	0.148	0.131
fr2/flowerbouquet_br.	0.726	0.726	0.733	0.724	0.727	0.731	<b>0.653</b>	0.731	0.703
fr2/large_no_loop	0.779	0.467	0.856	0.948	<b>0.354</b>	0.710	0.531	0.676	0.887
fr2/large_with_loop	<b>0.367</b>	0.405	0.376	0.371	0.377	0.380	0.411	0.383	3.598
fr2/metallic_sphere	1.698	1.233	1.815	1.399	1.527	1.936	1.287	<b>1.014</b>	1.099
fr2/metallic_sphere2	<b>0.701</b>	0.762	0.768	0.848	0.792	0.736	0.775	0.814	0.772
fr2/pioneer_360	0.215	0.213	0.230	0.211	0.211	<b>0.207</b>	0.219	0.210	0.213
fr2/pioneer_slam	0.361	0.362	<b>0.319</b>	0.333	0.345	0.351	0.357	0.326	0.367
fr2/pioneer_slam2	0.389	0.401	0.395	0.425	0.407	0.391	0.394	0.391	<b>0.381</b>
fr2/pioneer_slam3	0.342	0.527	0.332	0.341	0.345	<b>0.329</b>	0.334	0.331	0.511
Ortalama	0.533	0.495	0.559	0.536	0.492	0.544	0.510	<b>0.490</b>	0.762

RGB+Relative+128+2 (128 renk tonu nicelendirmesi ve 2 birim piksel uzaklığına göre hesaplanan RGB autocorrelogram ve relative uzaklık metriğinin birlikte kullanımı) ürettiği 48.2 cm ortalama hata ile en başarılı kombinasyondur. Ayrıca RGB+Relative+128+1 kombinasyonu da 48.3 cm ile en iyi sonuca çok yakın bir performans sergilemiştir. RGB ya da gri tonlamalı autocorrelogram kullanımı genellikle yakın sonuçlar üretmektedir (Bkz. Şekil 5.9 ve Şekil 5.10). Gri tonlamalı autocorrelogram ile elde edilen en düşük hata 49 cm olup (Gri+Relative+128+2) en iyi sonuca yakındır. Görüntünün 128 renk / gri tonu seviyesine nicelendirilmesi, 64 seviyeden genel olarak biraz daha başarılı sonuçlar üretmektedir. Aynı uzaklık metriği ve renk / gri tonu seviyesi kombinasyonu için, autocorrelogram hesaplanmasında 1 birim piksel uzaklığından 2 birime geçiş sonuçları üzerinde genellikle çok düşük bir etkiye sahiptir. Bu etki özellikle *fr2/metallic\_sphere* için daha büyüktür. Bu kayıt için Gri+Relative+128+1 ve Gri+Relative+128+2 kombinasyonları ile elde edilen RMS-ATE sonuçları arasındaki fark yaklaşık 92 cm'dir. *fr2/metallic\_sphere* için elde edilen hata oranları diğer kayıtlara göre daha yüksek olduğundan, farklı parametre kombinasyonları sonuçları önemli oranda etkilemektedir.



Şekil 5.9. RGB autocorrelogram benzerliği tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması



Şekil 5.10. Gri tonlmalı autocorrelogram benzerliği tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması

Elde edilen deneysel sonuçlara göre, döngü kapama adaylarının belirlenmesi için autocorrelogram benzerliğinin kullanılması, histogram benzerliği tekniğine göre daha başarılı sonuçlar üretmektedir. Autocorrelogram benzerliği ile elde edilen en yüksek ortalama kayma hatası 55.9 cm (Gri+Manhattan+64+2 kombinasyonu) olup, histogram benzerliği tekniğinin en başarılı kombinasyonu olan RGB+Chi-Square ile elde edilen ortalama hataya (54.5 cm) çok yakındır. Dolayısıyla, görüntüdeki renkler / gri tonu seviyeleri arasındaki uzamsal ilişkilerin

kullanılması, bu değerlerin genel dağılımının kullanımına göre doğruluğu daha fazla arttırmaktadır. RGB-D SLAM sistemi [44] ile elde edilen ortalama kayma hatası, autocorrelogram benzerliği tekniğinin en başarılı kombinasyonu (RGB+Relative+128+2) ile yaklaşık %37 oranında azaltılmıştır. Bu oran histogram benzerliği için yaklaşık %28'dir.

### 5.5.3. Histogram Benzerliği ile Nitelik Eşleştirmesi Deneyleri

Bu kesime kadarki deneylerde mekân tanıma için sadece genel tanımlayıcı benzerliği kullanılmıştır. Bu deney grubunda ise genel tanımlayıcı karşılaştırması ile birlikte yerel nitelikler de kullanılmıştır. Mevcut çerçeveye en benzer histograma sahip  $4 \times 8 = 32$  anahtar çerçeve grubu belirlendikten sonra, bu grup ile mevcut çerçeve arasında nitelik eşleştirmesi gerçekleştirilerek en fazla eşleşme üreten 8 döngü kapama adayı seçilmektedir. Çizelge 5.7 ve Çizelge 5.8 bu deneylere dair sırasıyla RGB ve gri tonlamalı histogram kullanımı için kayma hatası sonuçlarını içermektedir.

Çizelge 5.7. RGB histogram benzerliği ve nitelik eşleştirme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m)

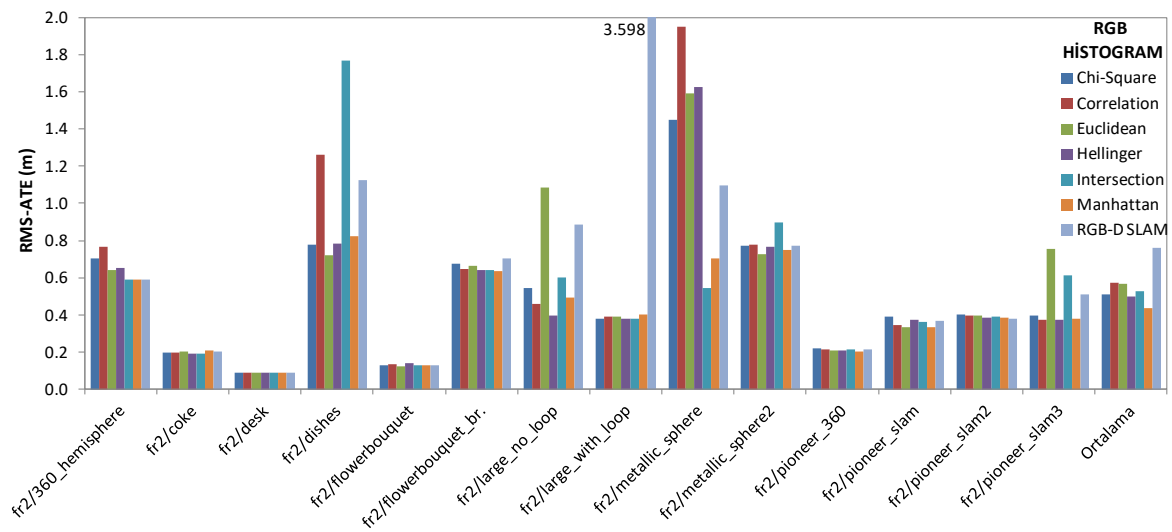
Uzaklık Metriği	Chi-Square	Correlation	Euclidean	Hellinger	Intersection	Manhattan	RGB-D SLAM
fr2/360_hemisphere	0.705	0.765	0.642	0.654	<b>0.591</b>	<b>0.591</b>	0.592
fr2/coke	0.196	0.200	0.203	0.193	<b>0.191</b>	0.207	0.205
fr2/desk	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>
fr2/dishes	0.776	1.259	<b>0.719</b>	0.785	1.766	0.825	1.126
fr2/flowerbouquet	0.127	0.138	<b>0.126</b>	0.138	0.127	0.129	0.131
fr2/flowerbouquet_br.	0.678	0.646	0.664	0.639	0.641	<b>0.638</b>	0.703
fr2/large_no_loop	0.545	0.458	1.085	<b>0.396</b>	0.602	0.494	0.887
fr2/large_with_loop	<b>0.378</b>	0.392	0.391	0.379	0.379	0.403	3.598
fr2/metallic_sphere	1.448	1.953	1.593	1.627	<b>0.547</b>	0.702	1.099
fr2/metallic_sphere2	0.772	0.778	<b>0.726</b>	0.769	0.898	0.747	0.772
fr2/pioneer_360	0.223	0.212	0.206	0.208	0.213	<b>0.205</b>	0.213
fr2/pioneer_slam	0.389	0.345	0.337	0.375	0.363	<b>0.335</b>	0.367
fr2/pioneer_slam2	0.400	0.399	0.398	0.387	0.391	0.386	<b>0.381</b>
fr2/pioneer_slam3	0.398	0.375	0.753	<b>0.372</b>	0.614	0.379	0.511
Ortalama	0.509	0.572	0.567	0.501	0.529	<b>0.438</b>	0.762

Ortalamada en başarılı sonuç 43.4 cm kayma hatası ile Gri+Correlation kombinasyonuna aittir. Ayrıca Gri+Intersection, RGB+Manhattan ve

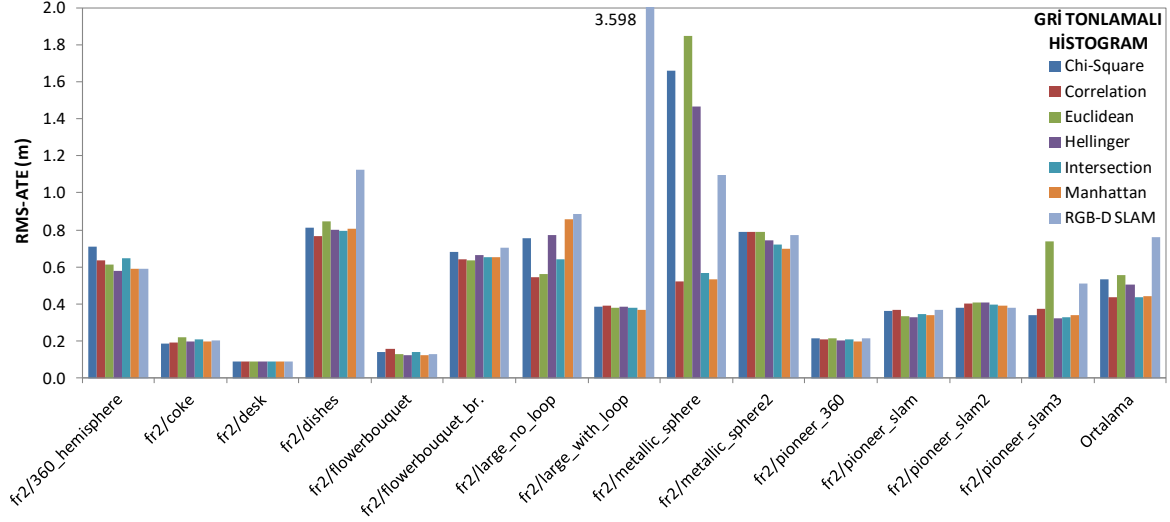
Gri+Manhattan kombinasyonları sırasıyla 43.7 cm, 43.8 cm ve 44.2 cm ortalama RMS-ATE ile en iyiye yakın performans sergilemişlerdir. Bölüm 5.5.1'deki sonuçlar ile karşılaştırma yapıldığında; Correlation, Intersection ve Manhattan metriklerinin daha olası adayları geniş bir grup halinde bulmakta daha başarılı oldukları ancak en benzer adayların ilk sıralarda seçilmesinde daha düşük performans sergiledikleri gözlenmektedir.

Çizelge 5.8. Gri tonlamalı histogram benzerliği ve nitelik eşleştirme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m)

Uzaklık Metriği	Chi-Square	Correlation	Euclidean	Hellinger	Intersection	Manhattan	RGB-D SLAM
fr2/360_hemisphere	0.707	0.636	0.614	<b>0.580</b>	0.645	0.591	0.592
fr2/coke	<b>0.188</b>	0.194	0.218	0.195	0.207	0.198	0.205
fr2/desk	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>
fr2/dishes	0.812	<b>0.767</b>	0.846	0.802	0.795	0.809	1.126
fr2/flowerbouquet	0.142	0.157	0.130	<b>0.122</b>	0.138	0.126	0.131
fr2/flowerbouquet_br.	0.682	0.643	<b>0.637</b>	0.663	0.652	0.652	0.703
fr2/large_no_loop	0.755	<b>0.545</b>	0.564	0.773	0.640	0.857	0.887
fr2/large_with_loop	0.383	0.389	0.377	0.382	0.378	<b>0.370</b>	3.598
fr2/metallic_sphere	1.660	<b>0.520</b>	1.849	1.469	0.570	0.533	1.099
fr2/metallic_sphere2	0.791	0.787	0.791	0.743	0.722	<b>0.698</b>	0.772
fr2/pioneer_360	0.213	0.209	0.215	0.204	0.212	<b>0.199</b>	0.213
fr2/pioneer_slam	0.360	0.365	0.332	<b>0.329</b>	0.343	0.337	0.367
fr2/pioneer_slam2	0.382	0.401	0.406	0.406	0.396	0.391	<b>0.381</b>
fr2/pioneer_slam3	0.337	0.372	0.736	<b>0.322</b>	0.331	0.339	0.511
Ortalama	0.536	<b>0.434</b>	0.557	0.506	0.437	0.442	0.762



Şekil 5.11. RGB histogram benzerliği ve nitelik eşleştirme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması



Şekil 5.12. Gri tonlamalı histogram benzerliği ve nitelik eşleştirme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması

Elde edilen sonuçlar, histogram karşılaştırmasından sonra nitelik eşleştirmesi uygulanmasının tüm kombinasyonlar için (özellikle gri tonlamalı histogram kullanılan) ortalama doğruluğu arttırdığını göstermektedir. Çizelge 5.3 ve Çizelge 5.4'teki sonuçlar göz önünde bulundurulduğunda, kayma hatasındaki ortalama azalma oranı gri tonlamalı histogram kombinasyonları için yaklaşık 19 cm (%28), RGB için ise 10 cm (%16) civarındadır. *fr2/metallic\_sphere* ve *fr2/dishes* kayıtları bu iyileştirme üzerinde büyük etkiye sahiptir (Bkz. Şekil 5.11 ve Şekil 5.12). Ayrıca en başarılı kombinasyonlar özelinde bu teknik autocorrelogram benzerliği tekniğinden daha başarılı sonuçlar üretmektedir. Bu teknikteki en etkili kombinasyon olan Gri+Correlation ile RGB-D SLAM [44]'in doğruluk oranı yaklaşık %43 arttırılmıştır. Bu oran, yalnızca genel tanımlayıcı kullanılan tekniklere kıyasla daha yüksek bir iyileştirme anlamına gelmektedir. Sonuç olarak, genel niteliklerin yanında nitelik eleştirme yoluyla yerel görüntü niteliklerinin de kullanımıyla yöntemin başarımının arttığı gözlenmektedir.

#### 5.5.4. Autocorrelogram Benzerliği ile Nitelik Eşleştirmesi Deneyleri

Bu kesimde ise autocorrelogram benzerliği ile nitelik eşleştirmesinin birlikte kullanımına dair deneysel sonuçlar incelenmiştir. Autocorrelogram benzerliğine göre mevcut çerçeveye en yakın  $4 \times 8 = 32$  anahtar çerçeve başlangıç grubu olarak seçilmekte ve bu gruptan mevcut çerçeveye en fazla sayıda anahtar nokta eşleşmesine sahip 8 çerçeve döngü kapama adayı olarak seçilmektedir. Bu



teknikle elde edilen sonuçlar sırasıyla RGB ve gri tonlamalı autocorrelogram kullanımı için Çizelge 5.9 ve Çizelge 5.10'da sunulmuştur.

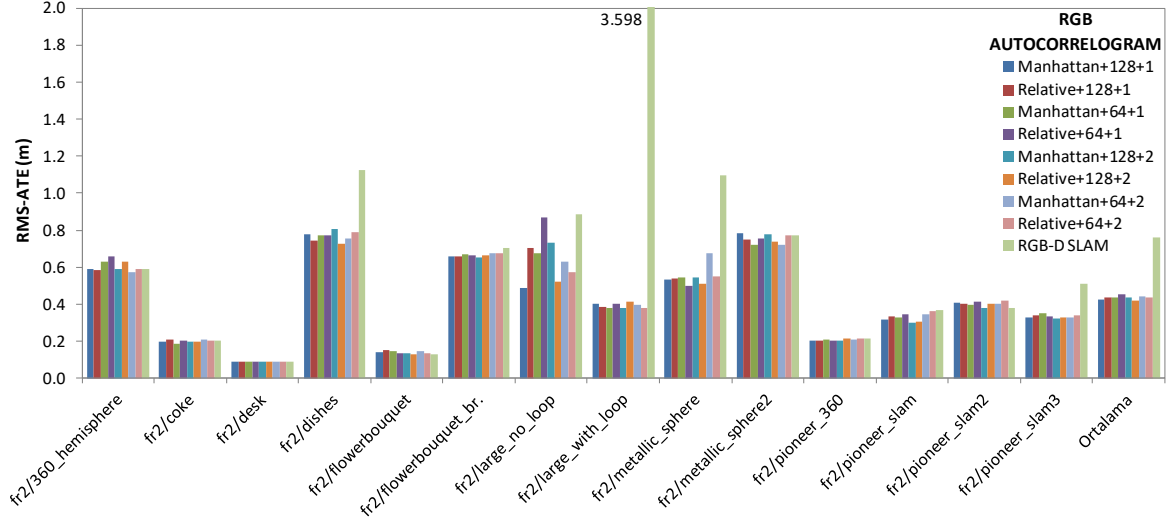
Çizelge 5.9. RGB autocorrelogram benzerliği ve nitelik eşleştirme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m)

Nicelendirme Faktörü	64				128				RGB-D SLAM
	1		2		1		2		
Piksel Uzaklığı	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	
fr2/360_hemisphere	0.631	0.656	<b>0.575</b>	0.588	0.591	0.585	0.590	0.630	0.592
fr2/coke	<b>0.187</b>	0.204	0.208	0.200	0.198	0.211	0.196	0.198	0.205
fr2/desk	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>
fr2/dishes	0.773	0.773	0.755	0.787	0.777	0.745	0.804	<b>0.728</b>	1.126
fr2/flowerbouquet	0.148	0.136	0.148	0.133	0.140	0.153	0.135	0.132	<b>0.131</b>
fr2/flowerbouquet_br.	0.667	0.661	0.677	0.675	0.656	0.658	<b>0.654</b>	0.665	0.703
fr2/large_no_loop	0.676	0.868	0.629	0.573	<b>0.489</b>	0.706	0.731	0.521	0.887
fr2/large_with_loop	0.379	0.404	0.394	<b>0.377</b>	0.404	0.384	0.381	0.415	3.598
fr2/metallic_sphere	0.547	<b>0.497</b>	0.678	0.552	0.535	0.540	0.542	0.508	1.099
fr2/metallic_sphere2	<b>0.721</b>	0.756	0.722	0.770	0.781	0.747	0.775	0.739	0.772
fr2/pioneer_360	0.209	<b>0.205</b>	0.206	0.212	0.206	0.206	<b>0.205</b>	0.215	0.213
fr2/pioneer_slam	0.329	0.345	0.346	0.361	0.318	0.335	<b>0.302</b>	0.303	0.367
fr2/pioneer_slam2	0.396	0.412	0.400	0.421	0.407	0.402	<b>0.379</b>	0.400	0.381
fr2/pioneer_slam3	0.350	0.332	0.330	0.341	0.329	0.339	<b>0.321</b>	0.329	0.511
Ortalama	0.436	0.453	0.440	0.434	0.423	0.436	0.436	<b>0.419</b>	0.762

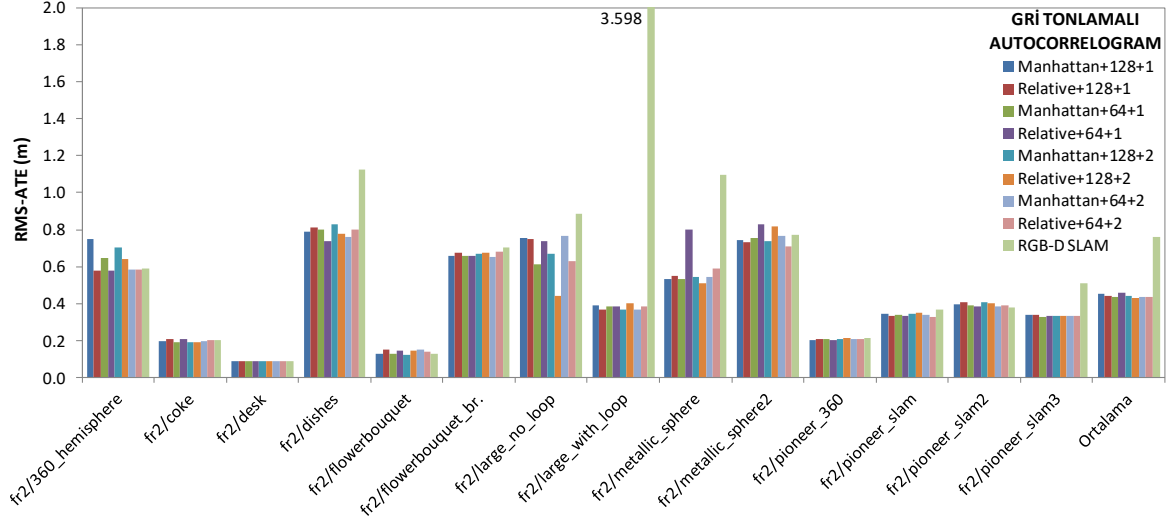
Çizelge 5.10. Gri tonlamalı autocorrelogram benzerliği ve nitelik eşleştirme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m)

Nicelendirme Faktörü	64				128				RGB-D SLAM
	1		2		1		2		
Piksel Uzaklığı	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	
fr2/360_hemisphere	0.649	<b>0.577</b>	0.584	0.585	0.748	0.579	0.705	0.640	0.592
fr2/coke	<b>0.189</b>	0.206	0.198	0.205	0.195	0.208	<b>0.189</b>	0.192	0.205
fr2/desk	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>
fr2/dishes	0.800	<b>0.740</b>	0.761	0.803	0.786	0.812	0.831	0.777	1.126
fr2/flowerbouquet	0.131	0.144	0.150	0.139	0.128	0.150	<b>0.123</b>	0.144	0.131
fr2/flowerbouquet_br.	0.658	0.659	<b>0.654</b>	0.679	0.658	0.673	0.668	0.678	0.703
fr2/large_no_loop	0.616	0.739	0.767	0.631	0.756	0.749	0.669	<b>0.441</b>	0.887
fr2/large_with_loop	0.385	0.386	<b>0.367</b>	0.383	0.389	0.371	0.369	0.402	3.598
fr2/metallic_sphere	0.535	0.801	0.545	0.593	0.536	0.550	0.544	<b>0.510</b>	1.099
fr2/metallic_sphere2	0.756	0.827	0.766	<b>0.711</b>	0.746	0.734	0.738	0.820	0.772
fr2/pioneer_360	0.210	0.204	0.209	0.207	<b>0.202</b>	0.209	0.211	0.213	0.213
fr2/pioneer_slam	0.340	0.336	0.341	<b>0.328</b>	0.347	0.334	0.343	0.354	0.367
fr2/pioneer_slam2	0.390	0.388	0.385	0.391	0.398	0.407	0.405	0.404	<b>0.381</b>
fr2/pioneer_slam3	<b>0.326</b>	0.333	0.331	0.331	0.338	0.341	0.336	0.332	0.511
Ortalama	0.434	0.459	0.439	0.434	0.451	0.443	0.444	<b>0.428</b>	0.762

Bir önceki kesimdeki sonuçlara benzer şekilde, nitelik eşleştirmesinin uygulanması yalnızca autocorrelogram kullanımına kıyasla tüm kombinasyonlar için ortalama kayma hatasının azalmasını sağlamıştır. Sadece genel tanımlayıcı kullanılan tekniklerde diğer kayıtlara göre daha fazla hata gözlenen *fr2/metallic\_sphere* kaydı için, bu teknikle daha gözle görülür bir iyileştirme görülmektedir. Ortalamada en düşük hata 41.9 cm ile RGB+Relative+128+2 kombinasyonu tarafından elde edilmiştir. Bunun yanında, değişik kombinasyonlarla elde edilen ortalama RMS-ATE değerlerinin birbirlerine çok yakın olduğu gözlenmiştir. Gözlenen en düşük ve en yüksek hata değerleri 41.9 cm - 45.9 cm aralığında olup, diğer tekniklere göre birbirlerine daha yakın sonuçlar elde edilmiştir. RGB ya da gri tonlamalı autocorrelogram kullanımı genel olarak yakın performans sergilemiştir (Bkz. Şekil 5.13 ve Şekil 5.14). Aynı autocorrelogram türü ile, görüntünün 64 ya da 128 renk / gri tonu seviyesine nicelendirilmesi benzer sonuçlar üretmektedir. Piksel uzaklığı parametresinin 1 birimden 2 birime çıkarılması doğruluk üzerinde ihmal edilebilir etkiye sahiptir.



Şekil 5.13. RGB autocorrelogram benzerliği ve nitelik eşleştirme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması



Şekil 5.14. Gri tonlamalı autocorrelogram benzerliği ve nitelik eşleştirme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması

Bu teknikte histogram benzerliğinin nitelik eşleştirmesi ile birlikte kullanımı karşılaştırıldığında, en başarılı kombinasyonlar özelinde iki tekniğin de yakın sonuçlar ürettiği gözlenmiştir. Genel durumda, RGB-D SLAM [44] ile elde edilen ortalama hata, autocorrelogram benzerliği ile nitelik eşleştirmesinin beraber kullanımıyla yaklaşık %45 oranında azaltılmıştır. RGB+Relative+128+2 kombinasyonu tarafından üretilen bu oran, tüm deneyler kapsamında elde edilen en yüksek doğruluk iyileştirmesidir.

### 5.5.5. İşlemsel Performans

Çizelge 5.11'de, önerilen teknikler ve RGB-D SLAM [44] sistemine dair çalışma süreleri yer almaktadır. İşlem zamanı hesaplamasına çizge optimizasyonu süresi de dâhil edilmiştir. Çerçeve başına işlem sürelerine dair standart sapma ortalamaları histogram benzerliği tekniği için 7 ms, autocorrelogram benzerliği tekniği için 6 ms, histogram benzerliği ile nitelik eşleştirme tekniği için 6.4 ms, autocorrelogram benzerliği ile nitelik eşleştirme tekniği için 6.8 ms, RGB-D SLAM [44] için ise 6.9 ms'dir.

Sonuçlar incelendiğinde, genel tanımlayıcı karşılaştırması sebebiyle gelen ekstra işlemsel maliyetin çok düşük olduğu görülmektedir. Histogram benzerliği tekniğinde her bir çerçeve için ortalama işlem süresi RGB kombinasyonlar için 172.4 ms, Gri tonlamalı kombinasyonlar için ise 172.9 ms'dir. RGB-D SLAM [44]

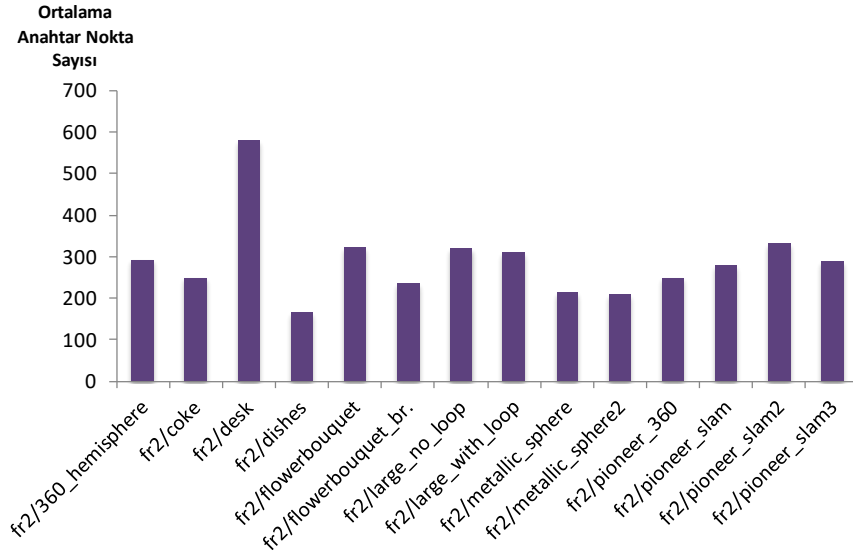
için elde edilen 171.3 ms değeriyle kıyaslandığında, histogram benzerliği tekniğinin neredeyse hiç ekstra yük getirmediği görülmektedir. Bu durum, histogram hesaplanması ve karşılaştırılması işlemlerinin çok hızlı şekilde gerçekleştirildiğini göstermektedir. Benzer şekilde, autocorrelogram benzerliği tekniği de düşük maliyet ile çalışmaktadır. Autocorrelogram hesaplanması 1 birimlik piksel uzaklığına göre gerçekleştirildiğinde ortalama ek maliyet çerçeve başına yaklaşık 6 ms, 2 birimlik uzaklık için ise yaklaşık 10 ms'dir. Piksel uzaklığı parametresinin değerinin artırılmasıyla birlikte, autoroccorelogram hesaplanması aşamasında her bir piksel için kontrol edilen piksel sayısı da arttığı için maliyet de yükselmektedir. Çizelge 5.11 incelendiğinde, histogram ve autocorrelogram benzerliği teknikleri ile bazı durumlarda (örneğin *fr2/coke* ve *fr2/metallic\_sphere2* kayıtları için) çerçeve başına işlem süresinin RGB-D SLAM [44] sistemine kıyasla düştüğü gözlenmektedir. Bu durum optimizasyon süresindeki dalgalanmalardan kaynaklanmaktadır.

Çizelge 5.11. Tüm uzaklık metrikleri için, önerilen teknikler ve RGB-D SLAM [44] ile elde edilen çerçeve başına işlem süreleri (ms)

Kayıt	Histogram		Autocorrelogram		Histogram + Nitelik Eşleştirme		Autocorrelogram + Nitelik Eşleştirme		RGB-D SLAM
	RGB	Gri Ton.	Pik. Uz. = 1	Pik. Uz. = 2	RGB	Gri Ton.	Pik. Uz. = 1	Pik. Uz. = 2	
fr2/360_hemisphere	156.7	151.9	156.4	161.6	199.6	189.3	189.9	193.8	154.1
fr2/coke	152.7	152.1	161.8	165.5	205.4	198.0	199.4	201.4	167.3
fr2/desk	325.5	376.6	372.8	383.3	586.9	584.7	578.4	586.2	348.6
fr2/dishes	143.1	143.3	140.6	143.7	149.0	147.0	149.0	151.9	135.4
fr2/flowerbouquet	206.9	213.4	214.1	217.9	291.0	283.6	283.5	289.2	202.6
fr2/flowerbouquet_br.	167.4	170.4	171.1	175.5	182.5	189.9	188.3	192.2	149.8
fr2/large_no_loop	147.4	144.6	148.6	150.5	218.4	212.8	211.2	214.7	143.9
fr2/large_with_loop	170.6	155.0	157.9	161.0	249.4	231.0	216.5	222.4	156.6
fr2/metallic_sphere	151.8	147.9	159.6	159.1	181.3	175.8	176.9	181.1	155.7
fr2/metallic_sphere2	132.1	122.8	134.1	137.4	158.4	156.8	160.1	163.7	136.4
fr2/pioneer_360	143.1	139.9	145.3	148.6	164.1	161.4	164.6	168.2	139.8
fr2/pioneer_slam	159.2	147.6	159.0	162.5	202.7	198.1	201.6	205.0	155.4
fr2/pioneer_slam2	202.1	202.6	206.1	208.6	243.4	241.2	246.7	247.9	198.9
fr2/pioneer_slam3	155.7	152.8	156.2	160.0	194.7	190.5	191.2	194.9	153.4
Ortalama	172.4	172.9	177.4	181.1	230.5	225.7	225.5	229.5	171.3

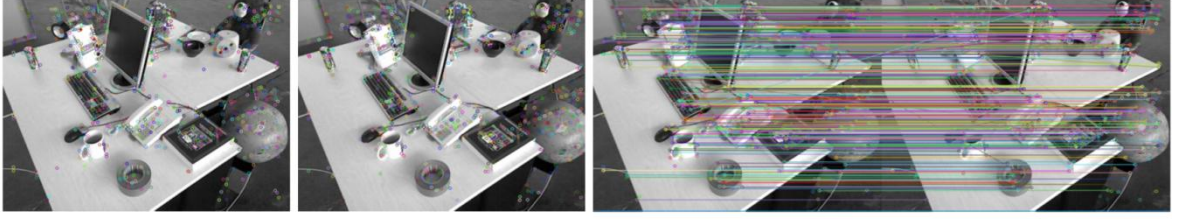
Elde edilen işlem süresi sonuçları, genel tanımlayıcı karşılaştırmasının ardından nitelik eşleştirmesi uygulanmasının maliyeti yaklaşık %33 oranında arttırdığını göstermektedir. Algılayıcıdan gelen her yeni çerçeve için anahtar noktaların tespit

edilmesi ve tanımlayıcıların oluşturulması işlemleri odometri tahmini aşamasında zaten gerçekleştirilmektedir. Önerilen mekân tanıma yaklaşımında hali hazırda mevcut olan tanımlayıcılar kullanıldığından, ekstra maliyet tanımlayıcıların karşılaştırılması işleminden kaynaklanmaktadır.

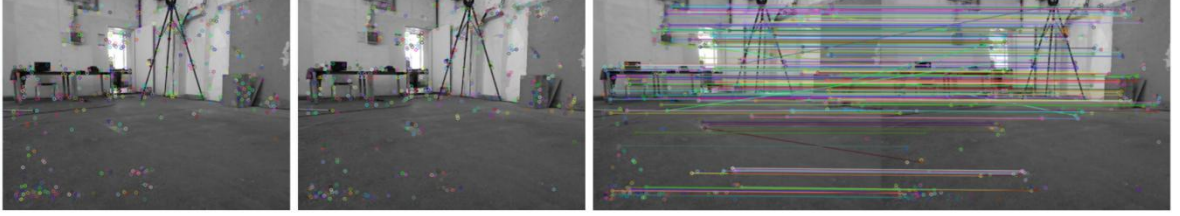


Şekil 5.15. CenSurE nitelik bulucusu ile tespit edilen, her bir kayıt için çerçeve başına ortalama anahtar nokta sayıları

*fr2/desk* görüntü kaydı için harcanan çerçeve başına işlem süresi diğer kayıtlara kıyasla çok daha fazladır çünkü bu kayıt anahtar nokta açısından diğerlerinden çok daha zengindir (Bkz. Şekil 5.15) ve algılayıcı hareketleri yeterli derecede stabildir. Dolayısıyla her yeni gelen çerçeve ile aday çerçeveler arasında hesaplanan geçerli dönüşüm sayısı daha hızlı artmakta ve bu da çizgedeki kenar sayısının daha hızlı yükselmesini sağlamaktadır. Bu durum tahmin edilen robot yolunun doğruluğunu arttırmakta ancak işlem hızını düşürmektedir. Buna benzer bir durum, *fr2/flowerbouquet* ve *fr2/pioneer\_slam2* kayıtlarında daha düşük etkiyle gözlenmektedir. Şekil 5.16'da, *fr2/desk* ve *fr2/pioneer\_360* kayıtları için art arda gelen çerçeveler arasındaki anahtar nokta eşleşmelerinden örnekler görülmektedir. Nitelik açısından daha zengin olan *fr2/desk* kaydında sırasıyla 659 ve 697 adet anahtar nokta içeren ardışık çerçeveler arasında 395 nokta eşleşmesi elde edilmiştir. *fr2/pioneer\_360* kaydı için ise sırasıyla 455 ve 442 anahtar nokta tespit edilen çerçeveler 231 eşleşme üretmişlerdir. *fr2/desk* görüntü kaydı için odometri tahmininde daha fazla avantaj sağlayan bu faktör, Şekil 5.16'dan da net olarak anlaşılmaktadır.



*fr2/desk*



*fr2/pioneer\_360*

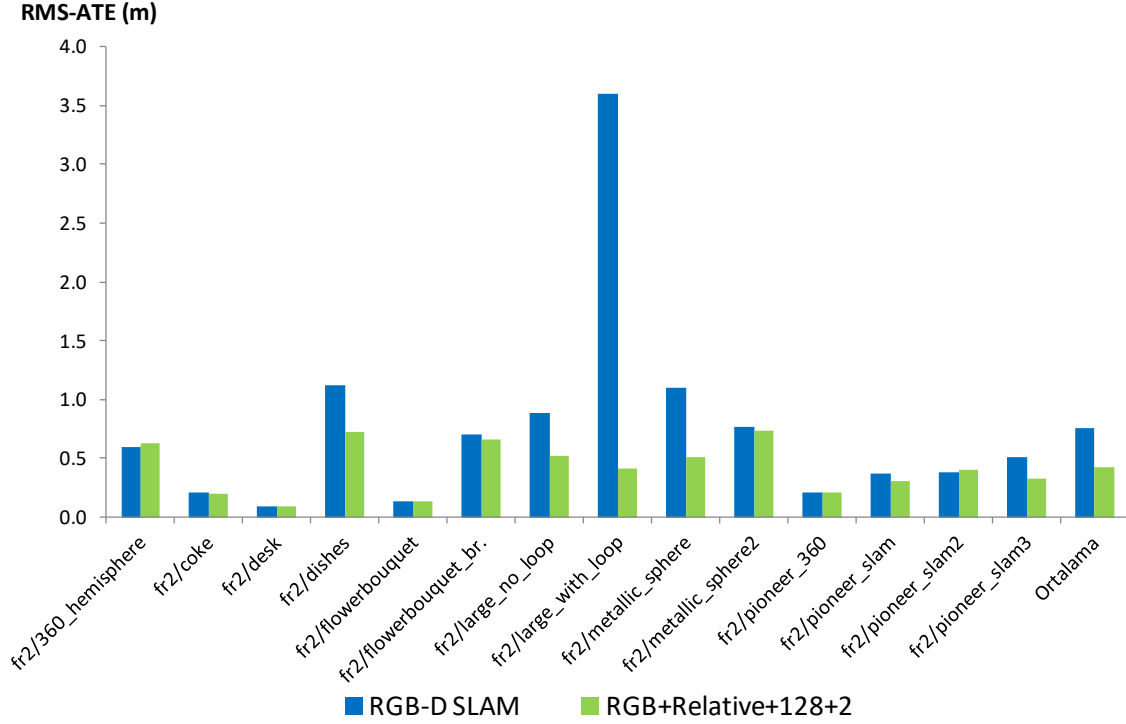
Ardışık Çerçevelerdeki Anahtar Noktalar      Çerçeveler Arasındaki Anahtar Nokta Eşleşmeleri

Şekil 5.16. *fr2/desk* ve *fr2/pioneer\_360* kayıtları için anahtar nokta tespiti ve eşleşme örnekleri

### 5.5.6. Tartışma

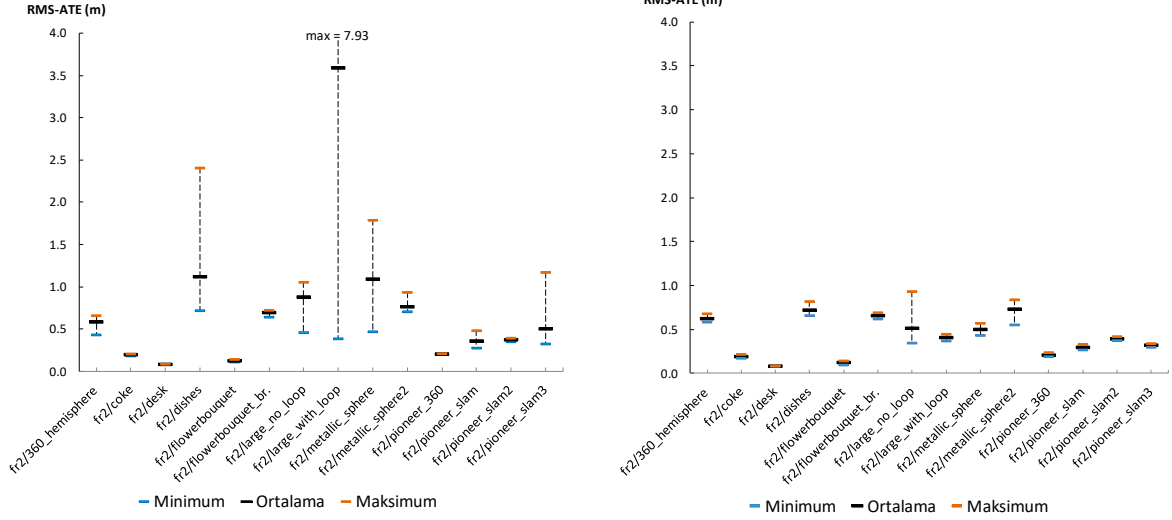
Deneysel sonuçlar, önerilen döngü kapama tespiti yönteminin farklı teknikler yoluyla uygulanmasının doğruluğu farklı seviyelerde arttırdığını göstermektedir. Yalnızca genel tanımlayıcı kullanılması hız açısından avantaj sağlamakta, yol doğruluğunu ise daha düşük oranlarda arttırmaktadır. Autocorrelogram, renk / gri tonu seviyesi değerlerinin uzamsal ilişkilerinden faydalanması sayesinde görüş açısı değişimlerine karşı daha dayanıklı performans sergilemekte ve histogramdan daha başarılı sonuçlar üretmektedir. Genel tanımlayıcı karşılaştırmasından sonra nitelik eşleştirmesi uygulanması, uzamsal bilgiden daha derinlemesine faydalanılmasını sağlayarak mekân tanıma başarımını arttırmaktadır. Bu işlemin işlemsel maliyeti genel tanımlayıcı karşılaştırmasından yüksektir. Dolayısıyla, önerilen iki-geçişli yöntem tüm anahtar çerçeveler yerine, genel tanımlayıcı benzerliğine göre seçilen bir anahtar çerçeve alt grubuna nitelik eşleştirmesi uygulayarak doğruluk ve verimlilik arasındaki dengeyi sağlamaktadır.

Şekil 5.17'de, RGB-D SLAM [44] ve önerilen döngü kapama tespiti yönteminin entegrasyonu (ortalamada en başarılı kombinasyon ile) arasındaki hata oranları karşılaştırması daha açık şekilde sunulmaktadır. Şekil 5.18 ise, 5 çalışma sonucunda elde edilen minimum, ortalama ve maksimum RMS-ATE değerlerini göstermektedir. Görüntü kayıtlarının büyük kısmında önerilen yöntem doğruluğu arttırmaktadır.



Şekil 5.17. RGB-D SLAM [44] ve önerilen yöntem (en başarılı kombinasyon olan RGB+Relative+128+2) arasında doğruluk karşılaştırması

Sistem başarımını etkileyen en önemli faktörlerden birisi tespit edilen anahtar nokta sayısıdır. *fr2/dishes*, *fr2/metallic\_sphere* ve *fr2/metallic\_sphere2* çerçeve başına ortalamada en az sayıda anahtar nokta tespit edilen üç kayıttır (Bkz. Şekil 5.15). RGB-D SLAM [44] bu kayıtlar için yüksek kayma hataları ile yol tahmini yapmaktadır çünkü çerçeveler arasında hesaplanan dönüşümler düşük sayıda modele uygun değer içermektedirler. Bu kayıtlardan *fr2/dishes* ve *fr2/metallic\_sphere* nispeten daha düşük öteleme hızı ile kaydedilmiştir (Bkz. Çizelge 5.1). Dolayısıyla bu iki kayıt için odometri tahmininde *fr2/metallic\_sphere2*'ye göre daha fazla sayıda direkt önceki çerçeve ile geçerli dönüşüm elde edilmektedir. Ancak bu durum, hesaplanan dönüşüm kalitesinin düşük olması sebebiyle hatayı arttırabilmektedir. Doğru döngü kapama tahmini bu durumda daha da önemli hale gelmektedir. RGB-D SLAM [44] döngü kapama adaylarını rastgele seçtiğinden dolayı, tahmin edilen yol hatalarına dair minimum-maksimum aralığı yüksektir. Şekil 5.18'de görüldüğü üzere önerilen yöntem bu kayıtlar için daha düşük hata ile, daha stabil yol tahminleri gerçekleştirmektedir.



(a) RGB-D SLAM

(b) RGB+Relative+128+2

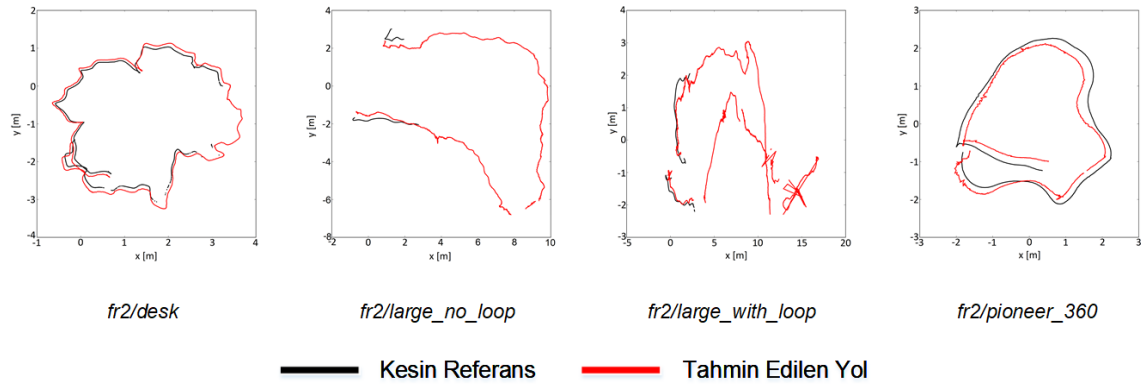
Şekil 5.18. RGB-D SLAM [44] ve en başarılı kombinasyon olan RGB+Relative+128+2 için detaylı mutlak yol hatası sonuçları

*fr2/pioneer\_360*, *fr2/pioneer\_slam*, *fr2/pioneer\_slam2* ve *fr2/pioneer\_slam3* kayıtları, RGB-D algılayıcının tekerlekli bir robot üzerine monte edilmesi yoluyla kaydedilmişlerdir. Dolayısıyla bu görüntü kayıtlarında algılayıcı yatay düzlemde stabil şekilde hareket ettirilmiştir. Bu durum odometri tahminini olumlu yönde etkilemiştir ancak kayıtların bazı kısımlarında derinlik verisinin elde edilememesi kayma hatasını arttıran önemli bir faktördür. Şekil 5.19'dan anlaşılacağı üzere bu kayıtlar için döngü kapama adayları başarılı şekilde tespit edilebilmektedir.

*fr2/large\_no\_loop* ve *fr2/large\_with\_loop* kayıtları diğerlerine göre çok daha geniş alanları kapsamaktadırlar. *fr2/large\_no\_loop* herhangi bir döngü kapama içermemektedir, dolayısıyla yanlış döngü kapama tespitlerinden kaçınılması çok önemlidir. Görüntülerin kaydedildiği endüstriyel salondaki tekrarlı yapılar yanlış eşleşmelere, dolayısıyla da yanlış döngü kapamalara sebep olabilmektedir. Buradan hareketle, nitelik eşleştirmesinde kullanılan uzaklık oranı yaklaşımının bu tarz yanlış eşleşmeleri elemekte başarılı olduğu söylenebilir. Geliştirilen döngü kapama tespiti yönteminin etkisini göstermeye en elverişli kayıt *fr2/large\_with\_loop*'tur. Bu kayıt, en sonda algılayıcının ilk baştaki konumuna döndüğü çok geniş bir yol ve büyük bir döngü kapama içermektedir. Bu kayıt diğerlerinden bu yönüyle ayrılmaktadır çünkü döngü kapama içermeyen



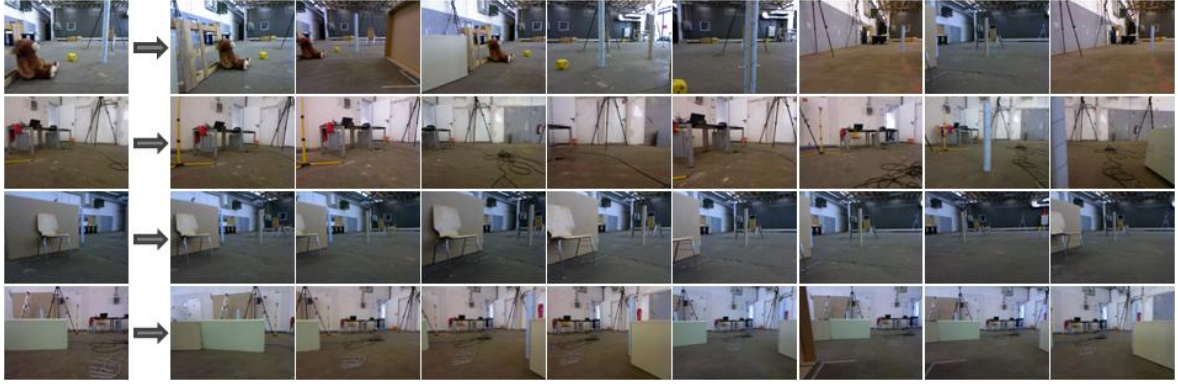
*fr2/large\_no\_loop* dışındaki diğer tüm kayıtlar nispeten daha kısa yollar ve çeşitli döngü kapamalar içermektedirler. *fr2/large\_with\_loop*'ta sadece bir döngü kapama mevcut olup, çok geniş olan bu döngünün tespit edilmesi haritanın doğruluğu açısından kritik öneme sahiptir. Şekil 5.17 ve Şekil 5.18, önerilen yöntemle döngü kapama adaylarının başarıyla tespit edildiğini ve kayma hatasının önemli oranda azaltıldığını göstermektedir. Ayrıca bu kayıtlar için tahmin edilen yollar Şekil 5.19'da yer almaktadır.



Şekil 5.19. En başarılı kombinasyon olan RGB+Relative+128+2 ile örnek kayıtlar için tahmin edilen yolun ve kesin referansın x-y düzlemine izdüşümü

*fr2/desk* diğer kayıtlardan çok daha fazla anahtar nokta içerdiğinden ve düşük hızda kaydedildiğinden dolayı, yüksek sayıda modele uygun değer ile hesaplanan çerçeveler arası dönüşümler bu kayıt için en düşük RMS-ATE değerlerinin elde edilmesini sağlamıştır. Ayrıca yine *fr2/coke* ve *fr2/flowerbouquet* de düşük hatayla odometri tahminine imkân veren, nispeten daha az zorlayıcı kayıtlardır. *fr2/flowerbouquet\_brownbackground* kaydında arka plan bir kartonla kapatılmıştır. Bu sebeple arka planda yer alan anahtar noktalar kullanılmadığından, *fr2/coke* ve *fr2/flowerbouquet*'e göre daha yüksek hata gözlenmiştir.

Şekil 5.20'de, en başarılı kombinasyon ile bazı örnek çerçeveler için elde edilen 8 döngü kapama adayı yer almaktadır. Şekilden de anlaşıldığı üzere döngü kapama adayları etkili şekilde tespit edilmektedir.



Şekil 5.20. Örnek çerçeveler için, en başarılı kombinasyon olan RGB+ Relative+128+2 kullanılarak belirlenen döngü kapama adayları (Adaylar hesaplanan benzerliğe göre soldan sağa doğru sıralanmıştır)

Döngü kapama tespitinin, oluşturulan haritanın tutarlılığı üzerindeki etkisi döngüyü kapayan çerçeveler arasında hesaplanan dönüşümün kalitesine bağlıdır. Eğer hesaplanan dönüşüm düşük sayıda modele uygun değer içeriyorsa ya da anahtar noktalar çerçevelerin küçük bölgelerinde toplanmışsa dönüşümün kalitesi olumsuz etkilenebilir ve doğruluk oranı düşebilir. Dolayısıyla, daha fazla sayıda döngü kapama tespit edilmesi harita doğruluğunu her zaman arttırmayabilir. Gerçekleştirilen deneylerde bu etki de gözlenmiştir. *fr2/360\_hemisphere*, RGB-D algılayıcının aynı noktada (öteleme yapılmadan) yukarı ve aşağı doğru hareket ettirilerek 360° döndürülmesiyle kaydedilmiş olup, veri kümesindeki en yüksek açısal hıza sahiptir. Görüntü kaydı içerisinde, endüstriyel salonun tavanını görüntüleyen çok sayıda çerçeve mevcuttur. Tavanın tekdüze yapısı, çerçeveler arasında yüksek kalitede dönüşüm hesaplanmasını zorlaştırmaktadır. Bu kayıt için; histogram benzerliği tekniğinde Gri+Chi-Square, autocorrelogram benzerliği tekniğinde RGB+Manhattan+128+1 ve autocorrelogram benzerliği ile nitelik eşleştirme tekniğinde Gri+Relative+64+1 kombinasyonları kullanılarak (Bkz. sırasıyla Çizelge 5.4, Çizelge 5.5 ve Çizelge 5.10) nispeten daha doğru sonuçlar elde edilmesine rağmen genel olarak kayma hatası azaltılamamıştır. Ayrıca, genel tanımlayıcı karşılaştırmasından sonra nitelik eşleştirmesi uygulanması daha fazla sayıda döngü kapama tespit edilmesine imkân vermekte ancak bazı durumlarda kayma hatasını arttırmaktadır. Örneğin *fr2/pioneer\_slam3* kaydı için elde edilen ortalama hata, RGB+Euclidean ve RGB+Intersection kombinasyonları için histogram karşılaştırmasından sonra nitelik eşleştirmesi gerçekleştirilmesi

durumunda sırasıyla yaklaşık 17 cm ve 21 cm artmaktadır (Bkz. Çizelge 5.3 ve Çizelge 5.7). Ayrıca *fr2/360\_hemisphere* görüntü kaydında, autocorrelogram benzerliği sonrası nitelik eşleştirmesi uygulanması sonucu ortalama hata Gri+Manhattan+128+1 kombinasyonu için yaklaşık 12 cm, Gri+Manhattan+128+2 kombinasyonunda ise 14 cm civarında artmaktadır (Bkz. Çizelge 5.6 ve Çizelge 5.10). Sonuç olarak; haritanın doğruluğu açısından en önemli nokta, çerçeveler arasında yüksek sayıda modele uygun değer içeren kaliteli dönüşümler üreten doğru döngü kapamaların tespit edilebilmesidir.

## 5.6. Değerlendirme

Bu bölümde SLAM sistemleri için bir döngü kapama tespiti yöntemi geliştirilmiştir. Yöntem genel ve yerel görüntü niteliklerini birlikte kullanmakta olup, nitelik tabanlı SLAM sistemlerine kolayca entegre edilebilmektedir. Mevcut görüntü çerçevesi ile önceki gözlemler arasındaki ilişkinin tespit edilebilmesi için genel nitelikler kullanılarak arama uzayı daraltılmakta, sonrasında ise yerel niteliklerin kullanımıyla döngü kapama adayları belirlenmektedir. Geliştirilen yöntem, döngü kapama adaylarını rastgele seçen bir SLAM sistemine entegre edilerek sistemin geniş ortamları etkin şekilde haritalayabilmesi sağlanmıştır.

Sistem başarımının değerlendirilmesi için, SLAM alanında yaygın olarak kullanılmakta olan bir veri kümesinde yer alan çok sayıda görüntü kaydı ile geniş kapsamlı deneyler gerçekleştirilmiştir. Farklı parametre kombinasyonları kullanılarak sonuçlara olan etkileri analiz edilmiştir. Deneysel sonuçlar neticesinde, geliştirilen yöntemin döngü kapama adaylarını başarıyla tespit ettiği ve kayma hatasını özellikle geniş kayıtlar için önemli oranda azalttığı gözlemlenmiştir. Orijinal RGB-D SLAM sistemi [44] ile karşılaştırma yapıldığında, geliştirilen yöntemin entegrasyonu sonucunda hatanın yaklaşık %45 oranında azaldığı görülmüştür. Ayrıca yerel nitelikler kullanılmadan yalnızca genel niteliklere göre seçim yapıldığında da çok düşük bir ekstra maliyetle başarımlar arttırılabilmektedir. Yerel niteliklerin kullanımı yaklaşık %33'lük bir ekstra maliyetle doğruluğu daha da arttırmaktadır.

Bu çalışmanın bir diğer önemli katkısı ise, autocorrelogram yapısının SLAM problemine mekân tanıma için adapte edilmesidir. Autocorrelogramlar dayanıklılıkları, az yer kaplamaları ve düşük maliyetle hesaplanabilmeleri

sebebiyle görüntü erişim sistemleri içerisinde etkin şekilde kullanılmaktadırlar. Bu çalışma, SLAM kapsamında döngü kapamaların tespit edilebilmesi için autocorrelogram kullanımının önemli bir seçenek olduğunu göstermiştir.

## 6. DÖNGÜ KAPAMA TESPİTİ YÖNTEMİNİN DAHA HIZLI VE ETKİLİ HALE GETİRİLMESİ

SLAM kapsamında döngü kapama tespitinin başarımı açısından, döngü kapama adaylarının doğru belirlenmesinin yanında hızlı şekilde seçilmesi de kritik öneme sahiptir. Robotun ortamdaki gezinimiyle birlikte sürekli genişlemekte olan arama uzayında, doğru aday seçiminin zorluğu yanında bir diğer büyük problem ise arama işleminin yüksek maliyetidir. Özellikle geniş ortam haritalamasında uzak döngü kapamaların önemi çok daha fazladır, ancak hızlı ve doğru tespit edilmeleri bir o kadar zordur. Bu açıdan bakıldığında, geniş ortamlarda gerçek zamanlı çalışması gereken bir SLAM sistemi için hızlı ve doğru döngü kapama tespiti olmazsa olmaz bir gerekliliktir.

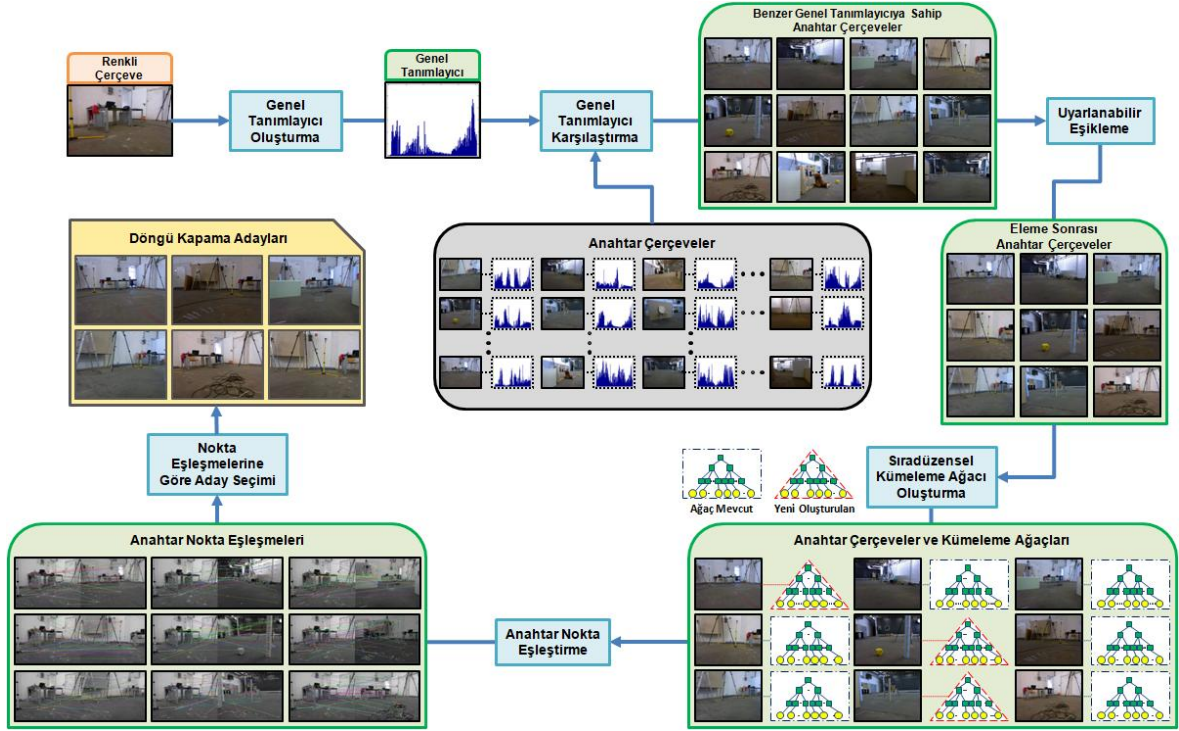
Bu çalışmada [87], önceki kesimde önerilen döngü kapama tespiti yöntemi iki yeni teknik ile genişletilerek daha hızlı ve etkili hale getirilmiştir. Sıradüzensel kümeleme (*hierarchical clustering*) ve uyarlanabilir eşikleme (*adaptive thresholding*) teknikleri kullanılarak yöntemin verimliliğinin artırılmasının yanında başarımının da korunması hedeflemiştir. Nitelik eşleştirme işleminde sıradüzensel kümeleme yaklaşımı benimsenerek işlem maliyeti azaltılmakta ve *brute-force* yönteme yakın eşleştirme başarısı elde edilebilmektedir. Uyarlanabilir eşikleme ise modele aykırı döngü kapama adaylarının genel tanımlayıcı benzerliğine göre elenmesini sağlamaktadır. Dolayısıyla yöntemin hızını arttırmakta ve doğruluğuna katkı sağlamaktadır.

Yöntem genişletilmiş haliyle RGB-D SLAM sistemine [44] entegre edilmiş ve özellikle geniş ölçekli ortam haritalamasındaki performansı gerçekleştirilen deneylerle ölçülmüştür.

### 6.1. Önerilen Genişletilmiş Döngü Kapama Tespiti Yöntemi

Önceki bölümde (Bkz. Bölüm 5) de belirtildiği üzere yöntem genel ve yerel görüntü niteliklerini birlikte kullanmakta, nitelik tabanlı SLAM sistemlerine kolayca entegre edilebilmektedir. Yöntem genişletilmiş haliyle birbirini tamamlayan 3 ana bileşenden oluşmaktadır; genel tanımlayıcı karşılaştırması, uyarlanabilir eşikleme ve sıradüzensel kümeleme. Yöntemin genel akışında; genel tanımlayıcı benzerliğine göre belirlenen başlangıç aday grubu üzerinde uyarlanabilir eşikleme uygulanarak modele aykırı adaylar filtrelenmekte, sonrasında ise mevcut çerçeve

ile grup arasında sıradüzensel kümeleme tabanlı nitelik eşleştirmesi gerçekleştirilerek döngü kapama adayları belirlenmektedir. Yöntemin genel işleyişi Şekil 6.1'de görülmektedir.



Şekil 6.1. Önerilen genişletilmiş döngü kapama tespiti yönteminin genel görünümü

### 6.1.1. Sıradüzensel Kümeleme ile Mekân Tanıma

Geliştirilen döngü kapama tespiti yönteminin ilk versiyonunda uygulandığı şekliyle (Bkz. Bölüm 5.3.2) *brute-force* nitelik eşleştirmesi, özellikle çerçeveler anahtar nokta açısından zengin olduğunda işlemsel olarak yüksek maliyet getirmektedir. Dolayısıyla *brute-force* yaklaşıma alternatif olarak, Muja ve Lowe [88] tarafından önerilen algoritma kullanılmıştır. Bu algorithmada arama uzayı sıradüzensel olarak parçalara ayrılarak daha hızlı şekilde anahtar nokta eşleştirmesi hedeflenmektedir. Noktalar art arda kümelenerek bir sıradüzensel kümeleme ağacı (*hierarchical clustering tree*) oluşturulmaktadır. Bu ağaçta ara düğümler küme merkezlerini ifade etmekte, yaprak düğümler ise noktaları içermektedir. Oluşturulacak küme sayısı *dallanma\_faktörü* parametresi ile belirlenmekte ve küme merkezleri noktalar arasından rastgele seçilmektedir. Algoritmadaki bir diğer parametre ise, bir kümede yer alabilecek maksimum nokta sayısını belirleyen *maksimum\_yaprak\_büyüküğü*'dür.

---

```

Input : tüm noktaları içeren küme  $D$ 
Output: sıradüzensel kümeleme ağacı
Parameters:  $dallanma\_faktörü$ ,  $maksimum\_yaprak\_büyüklüğü$ 
if  $|D| < maksimum\_yaprak\_büyüklüğü$  then
  |  $D$ 'deki noktalar ile yaprak düğüm oluştur
else
  |  $R \leftarrow D$  kümesinden rastgele  $dallanma\_faktörü$  adet noktayı küme merkezleri olarak seç
  |  $C \leftarrow D$ 'deki noktaları en yakın merkezler etrafında kümele
  | for her bir küme  $C_i \in C$  do
  |   |  $R_i$  merkezli ara düğümü oluştur
  |   | algoritmayı  $C_i$ 'deki noktalara özyineli olarak uygula
  | end
end
return sıradüzensel kümeleme ağacı

```

---

Şekil 6.2. Sıradüzensel kümeleme ağacı oluşturma

Sıradüzensel kümeleme ağacının oluşturulmasında (Bkz. Şekil 6.2) algoritma tüm noktalar ile işleme başlamaktadır. Küme merkezleri rastgele seçilmekte, bu merkezler için ara düğümler oluşturulmakta ve her bir nokta en yakın küme ile ilişkilendirilmektedir. Sonrasında ise,  $maksimum\_yaprak\_büyüklüğü$ 'nden daha az sayıda nokta içeren her bir küme için, o kümede yer alan noktalara sahip bir yaprak düğüm oluşturulmaktadır. Diğer kümeler ise aynı yaklaşımla tekrar bölünmektedir. Bu işlem, her bir küme  $maksimum\_yaprak\_büyüklüğü$ 'nden daha az sayıda nokta içerinceye kadar özyineli olarak uygulanmaktadır. Bu şekilde, noktaları kapsayan kümeleme ağacı sıradüzensel şekilde oluşturulmaktadır.

Noktalar arasında eşleştirme işlemi, sıradüzensel kümeleme ağacının özyineli olarak dolaşılması ile gerçekleştirilmektedir (Bkz. Şekil 6.3). Herhangi bir arama noktasına (*query point*) karşılık gelen eşleşmeyi bulmak için, ağaç dolaşılırken her bir adımda arama noktasına en yakın düğüm seçilmektedir. Yaprak düğüme ulaşıldıktan sonra, o kümede yer alan noktalar eşleşme için kontrol edilmektedir. Ağaç dolaşımı sırasında ele alınmayan düğümler ise bir öncelik kuyruğunda (*priority queue*) tutulmaktadır. Eşleşme arama işlemi, bu kuyruktaki en yakın düğüm seçilerek oradan dolaşıma devam edilmesi yoluyla genişletilebilmektedir. Algoritma, birden fazla ağaç oluşturulması ve bu ağaçlarda paralel şekilde arama yapılmasına da imkân vermektedir. Birden fazla ağaç kullanımı arama keskinliğini arttırmaktadır ancak gerek ağaç oluşturma süresinin gerekse kullanılan bellek miktarının yükselmesi sebebiyle ekstra işlemsel yük getirmektedir.



---

```

Input : sıradüzensel kümeleme ağacı  $T$ , arama noktası  $Q$ 
Output: arama noktasının yaklaşık en yakın  $K$  komşusu
Parameters: kontrol edilen maksimum nokta sayısı  $N_{max}$ 

 $N \leftarrow 0$  {kontrol edilen nokta sayısını tutar}
 $PQ \leftarrow$  ele alınmayan düğümler için boş öncelik kuyruğu
 $S \leftarrow$  bulunan komşular için boş öncelik kuyruğu
call AgaciDolas ( $T, PQ, S$ )
while  $|PQ| > 0$  and  $N < N_{max}$  do
  |  $F \leftarrow PQ$  kuyruğundaki en üst düğüm
  | call AgaciDolas ( $F, PQ, S$ )
end
return  $S$ 'de en üstte yer alan  $K$  nokta

Procedure AgaciDolas( $F, PQ, S$ )
  | if  $F$  yaprak düğüm then
  | |  $F$ 'deki tüm noktaları  $S$ 'ye ekle
  | |  $N \leftarrow N + |F|$ 
  | else
  | |  $C \leftarrow F$ 'nin alt düğümleri
  | |  $C_q \leftarrow C$ 'de yer alan, arama noktası  $Q$ 'ya en yakın düğüm
  | |  $C_p \leftarrow C - C_q$ 
  | |  $C_p$ 'deki tüm düğümleri  $PQ$ 'ya ekle
  | | call AgaciDolas ( $C_q, PQ, S$ )
  | end
end

```

---

Şekil 6.3. Sıradüzensel kümeleme ağacında arama

Şekil 6.4'teki algoritmada sıradüzensel kümeleme tekniği ile döngü kapama adaylarının seçimi görülmektedir. Bu teknikte, her bir anahtar çerçeve için bir kümeleme ağacı kullanılmaktadır. Öncelikle, genel tanımlayıcı benzerliğine göre  $gruplama\_faktörü \times aday\_sayısı$  büyüklüğündeki başlangıç aday grubu elde edilmektedir. Daha sonra ise bu gruptaki her bir anahtar çerçevenin kendi kümeleme ağacına sahip olup olmadığı kontrol edilmektedir. Ağaca sahip olmayan anahtar çerçeveler için, odometri tahmini aşamasında zaten elde edilmiş olan anahtar noktalar kullanılarak kümeleme ağacı oluşturulmaktadır. Herhangi bir anahtar çerçeve için bir kez kümeleme ağacı oluşturulduktan sonra, bu çerçevenin ileride dâhil olacağı tüm nitelik eşleştirme işlemlerinde aynı ağaç kullanılmaktadır. Dolayısıyla tüm haritalama süreci boyunca, her bir anahtar çerçeve için ağaç oluşturma işlemi sadece bir kez gerçekleştirilmektedir. Kümeleme ağaçlarının oluşturulmasından sonra, mevcut çerçeve ile başlangıç grubundaki her bir anahtar çerçeve arasında, mevcut çerçevedeki her bir anahtar noktanın arama noktası olarak uygulanması yoluyla nitelik eşleştirilmesi gerçekleştirilmektedir. Bu aşamada, *brute-force* nitelik eşleştirme tekniğinde de kullanılan uzaklık oranı yaklaşımı [24]



kullanılarak yanlış eşleşmeler elenmektedir (Bkz. Bölüm 5.3.2 ve Eşitlik 5.11). Nitelik eşleştirme sonuçlarına göre, mevcut çerçeve ile en fazla sayıda eşleşme üreten *aday\_sayısı* kadar anahtar çerçeve döngü kapama adayı olarak belirlenmektedir.

---

```

Input : mevcut çerçeve  $C$ 
          genel tanımlayıcı karşılaştırması ile elde edilen başlangıç anahtar çerçeve grubu  $G$ 
Output: döngü kapama adayları
Parameters: oran_ eşiği, aday_sayısı
for her bir anahtar çerçeve  $G_f \in G$  do
  | if  $G_f$  kümeleme ağacına sahip değil then
  | |  $D_{G_f} \leftarrow$  anahtar çerçeve  $G_f$ 'ye ait nitelik tanımlayıcıları
  | |  $T_{G_f} \leftarrow D_{G_f}$ 'yi kullanarak anahtar çerçeve  $G_f$  için kümeleme ağacı oluştur
  | end
end
 $K \leftarrow$  mevcut çerçeve  $C$ 'ye ait nitelik tanımlayıcıları
for her bir anahtar çerçeve  $G_f \in G$  do
  | for her bir nitelik tanımlayıcısı  $k \in K$  do
  | |  $T_{G_f}$ 'de arama yap ve  $k$ 'nın Hamming uzaklığına göre en yakın iki komşusunu bul
  | |  $c \leftarrow G_f$ 'deki en yakın anahtar noktaya olan uzaklık
  | |  $sc \leftarrow G_f$ 'deki ikinci en yakın anahtar noktaya olan uzaklık
  | | if  $c/sc > \text{oran\_eşiği}$  then
  | | |  $M_{G_f} \leftarrow$  eşleşmeyi  $G_f$ 'nin kabul edilen eşleşmeler kümesine ekle
  | | end
  | end
  |  $L \leftarrow$  anahtar çerçeve  $G_f$ 'yi,  $M_{G_f}$ 'nin büyüklüğüne göre azalan sırada listeye ekle
end
return  $L$ 'deki ilk aday_sayısı adet anahtar çerçeve

```

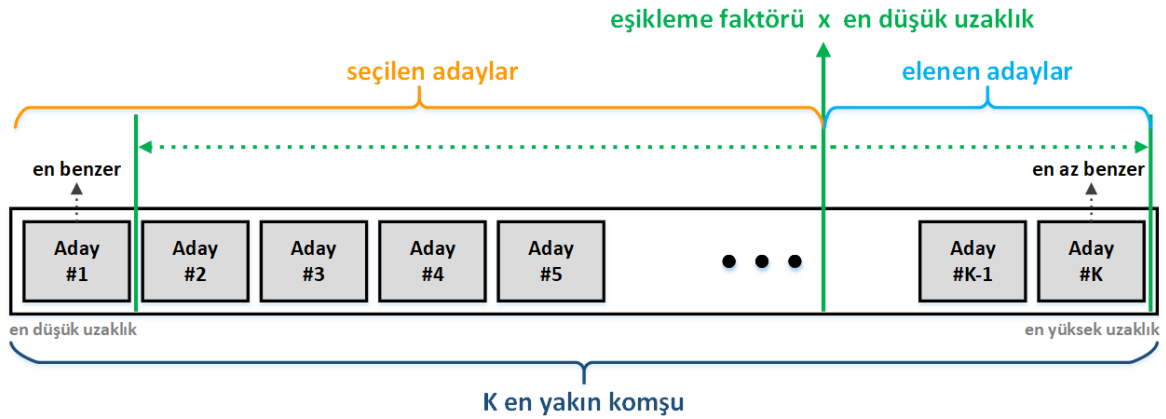
---

Şekil 6.4. Sıradüzensel kümeleme ile aday seçimi

### 6.1.2. Uyarlanabilir Eşikleme ile Mekân Tanıma

Genel görüntü niteliklerinin yanında yerel niteliklerin de kullanıldığı tekniklerde (Bkz. Bölüm 5.3.2 ve Bölüm 6.1.1) başlangıç aday grubu, genel tanımlayıcı benzerliğine göre *gruplama\_faktörü*  $\times$  *aday\_sayısı* adet anahtar çerçeveden oluşmaktadır. Bazı durumlarda başlangıç grubu mevcut çerçeveye yeterince benzer olmayan anahtar çerçeveler içerebilmektedir. Dolayısıyla başlangıç grubu olarak sabit sayıda anahtar çerçevenin seçilmesi yerine, mevcut çerçeveye yeterli derecede benzer anahtar çerçeveler belirlenebilir. Bu şekilde, anahtar nokta eşleştirme işleminden önce başlangıç grubundan bazı modele aykırı çerçeveler elenebilir. Bu noktadan hareketle, başlangıç grubundaki her bir anahtar çerçeve ile mevcut çerçeve arasındaki benzerliğe bakılarak, belirli bir eşik değerden düşük benzerliğe sahip anahtar çerçeveler elenmektedir. Uygulanacak eşik değer,

başlangıç grubundaki mevcut çerçeveye en benzer adayın benzerlik sonucunun belirli bir katsayı ile çarpılması yoluyla dinamik olarak hesaplanmaktadır. Daha açık olarak ifade etmek gerekirse, benzerlik eşiği şu şekilde hesaplanmaktadır;  $gruptaki\_en\_iyi\_sonuç \times eşikleme\_faktörü$ . Her bir uzaklık metriği için, tüm çerçeveler üzerindeki genel benzerlik dağılımı incelenerek farklı bir  $eşikleme\_faktörü$  uygulanmaktadır. Benzerlik eşiğinin değeri her bir döngü kapama araması için dinamik olarak belirlenmekte ve algılayıcıdan gelen her yeni çerçeve için uyarlanabilir şekilde değişmektedir. Dolayısıyla, modele aykırı adayların elenmesini sağlayan bu teknik uyarlanabilir eşikleme şeklinde adlandırılmıştır (Bkz. Şekil 6.5).



Şekil 6.5. Uyarlanabilir eşikleme mekanizması

Başlangıç grubundaki modele aykırı adayların uyarlanabilir eşikleme ile elenmesinden sonra, gruptaki anahtar çerçeveler ile nitelik eşleştirme aşamasına geçilmektedir. Sahip olduğu doğruluk ve işlemsel verimlilik sebebiyle, nitelik eşleştirmesi için sıradüzensel kümeleme yöntemi kullanılmaktadır. Mevcut çerçeve ile gruptaki adaylar arasında anahtar noktalar eşleştirildikten sonra, eşleşme sayısına göre  $aday\_sayısı$  adet anahtar çerçeve döngü kapama adayı olarak seçilmektedir.

Bölüm 5'tekine benzer şekilde, önerilen genişletilmiş döngü kapama tespiti yöntemi RGB-D SLAM sistemine [44] entegre edilmiş ve döngü kapama adaylarının rastgele seçilmesi yerine bu yöntemle belirlenmesi sağlanmıştır (Bkz. Şekil 5.2). Sıradaki kesimde, yöntemin genişletildiği teknikler kullanılarak elde edilen sonuçlar sunulmuştur.

## 6.2. Deneysel Kurulum

Gerçekleştirilen deneylerde, geniş ortam haritalamasındaki başarımın değerlendirilmesi için Bölüm 5'te olduğu gibi yine TUM RGB-D Benchmark veri kümesindeki *fr2* kayıtları kullanılmıştır. Bu kayıtlarla özellikle uzak döngü kapamaların etkin ve hızlı şekilde tespit edilip edilemediği kestirilmeye çalışılmıştır. *fr2* kayıtları ile ilgili ayrıntılı bilgiler için Çizelge 5.1 incelenebilir.

Odometri tahmini aşamasında ve döngü kapama tespiti için, nitelik bulucu ve tanımlayıcı olarak yine CenSurE ve FREAK ikilisi kullanılmıştır. Önceki kesimlerde de belirtildiği üzere, bu ikili doğruluk ve hız birlikte değerlendirildiğinde en iyi performansa sahiptir (Bkz. Bölüm 4).

Çizelge 6.1'de deneylerde uygulanan parametre değerleri görülmektedir. Parametrelere verilen değerler Bölüm 5'tekilerle aynı olup, geliştirilen yeni tekniklere ait bazı yeni parametreler sisteme eklenmiştir. Önceki deneylerde olduğu gibi; RANSAC yineleme sayısı 250, çerçeve başına anahtar nokta sayısı 700, *oran\_eşiği* ise 0.8 olarak uygulanmış ve nitelik eşleştirmesinde Hamming mesafesi kullanılmıştır. Histogramlardaki kanallar 32 bin üzerinden oluşturulmuş, autocorrelogramlar için ise 64 ve 128 nicelendirme faktörü ile 1 ve 2 piksel uzaklığı uygulanmıştır. Sıradüzensel kümeleme ağaçları ise *dallanma\_faktörü* için 32 ve *maksimum\_yaprak\_büyüklüğü* için 100 değerleri kullanılarak oluşturulmuştur.

Aday çerçeveler için uygulanan parametre değerleri de yine benzer şekilde  $n = 8$  direkt önceki,  $k = 20$  komşu ve  $l = 8$  döngü kapama çerçevesi şeklinde belirlenmiştir. Başlangıç aday grubu, *gruplama\_faktörü* için 4 değerinin uygulanmasıyla, genel tanımlayıcı benzerliğine göre  $4 \times 8 = 32$  anahtar çerçeveden oluşmaktadır. Anahtar nokta eşleştirmesi sonrasında ise  $aday\_sayısı = 8$  adet döngü kapama adayı belirlenmektedir.

Çizelge 6.1. Deneylerde uygulanan parametre değerleri

Parametre	Değer
<b><i>Dönüşüm Parametreleri</i></b>	
Maksimum RANSAC yineleme sayısı	250
Maksimum anahtar nokta sayısı	700
<i>oran_eşiği</i>	0.8
<b><i>Mekan Tanıma Parametreleri</i></b>	
Histogram bin sayısı	32
Nicelendirme faktörü	64,128
Piksel mesafesi	1,2
<i>dallanma_faktörü</i>	32
<i>maksimum_yaprak_büyüklüğü</i>	100
<i>gruplama_faktörü</i>	4
<b><i>Aday Parametreleri</i></b>	
Direkt önceki aday sayısı ( <i>n</i> )	8
Komşu aday sayısı ( <i>k</i> )	20
Döngü kapama adayı sayısı ( <i>l</i> )	8

Bu kesimdeki deneylerde, önceki deneylerdekine ek olarak kullanılan parametreler için uygulanan değerler benzer şekilde çeşitli denemeler ile elde edilen analiz sonuçlarına göre belirlenmiştir. Deneyler önceki kesimlerdekiyle aynı platform (Ubuntu 12.04 işletim sistemi kullanan, Intel Core i7-2600 CPU 3.40GHz işlemci ve 8GB RAM içeren bir bilgisayar) üzerinde 5'er defa gerçekleştirilmiş ve elde edilen ortalama sonuçlar nihai sonuç olarak listelenmiştir.

### 6.3. Sonuçlar

Bu kesimde, önerilen genişletilmiş döngü kapama tespiti yönteminin, içerdiği sıradüzensel kümeleme ve uyarlanabilir eşikleme bileşenlerinin histogram veya autocorrelogram ile birlikte kullanılmasıyla elde edilen sonuçlar doğruluk ve hız açısından incelenmiştir. İlk iki grup deneylerde, genel tanımlayıcı karşılaştırması ile belirlenen 32 anahtar çerçevelik başlangıç grubu üzerinde nitelik eşleştirmesi gerçekleştirilmiştir. Ancak nitelik eşleştirmesi aşamasında Bölüm 5'teki gibi *brute-force* yaklaşımın benimsenmesi yerine, yönteme eklenen yeni bileşen olan sıradüzensel kümeleme tabanlı yaklaşım kullanılmıştır. Sonraki grup deneylerde ise uyarlanabilir eşikleme bileşeni de aktif hale getirilerek, genişletilen döngü kapama tespiti yöntemi tüm bileşenleri ile birlikte test edilmiştir.

### 6.3.1. Histogram Benzerliđi ile Sıradüzensel Kümeleme Deneyleri

Bu deneylerde, genel tanımlayıcı olarak histogram benzerliđine göre  $4 \times 8 = 32$  anahtar çerçeveden oluşan başlangıç grubu ile mevcut çerçeve arasında sıradüzensel kümeleme tabanlı nitelik eşleřtirmesi uygulanmıř ve en fazla eşleşmeye sahip 8 anahtar çerçeve döngü kapama adayı olarak seçilmiřtir. Deneylere dair mutlak yol hatası sonuçları, Çizelge 6.2 ve Çizelge 6.3'te sırasıyla RGB ve gri tonlamalı histogram için sunulmuřtur.

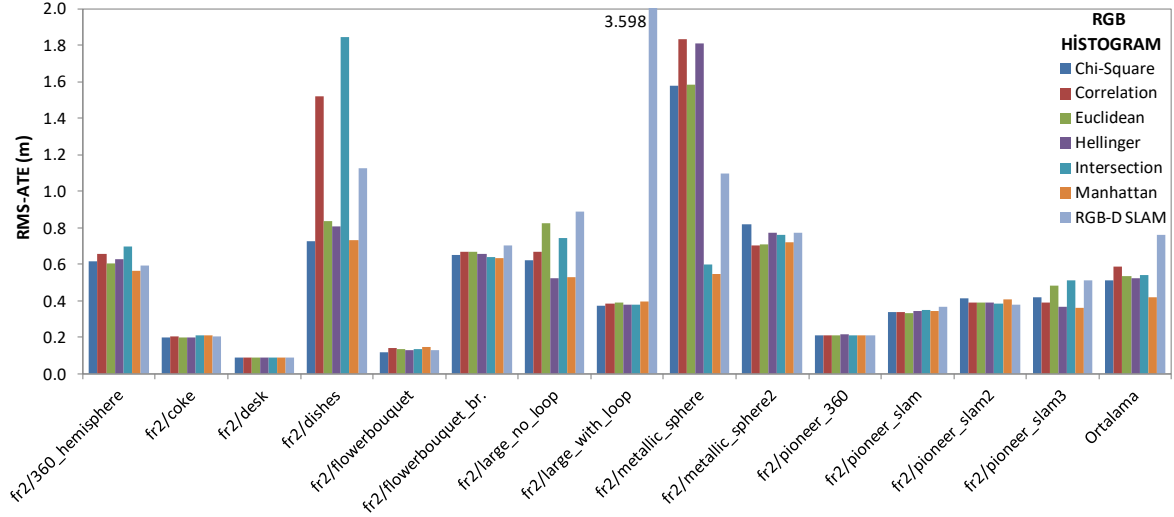
Çizelge 6.2. RGB histogram benzerliđi ve sıradüzensel kümeleme tekniđi için mutlak yol hatası (RMS-ATE) sonuçları (m)

Uzaklık Metriđi	Chi-Square	Correlation	Euclidean	Hellinger	Intersection	Manhattan	RGB-D SLAM
fr2/360_hemisphere	0.617	0.654	0.604	0.630	0.700	<b>0.566</b>	0.592
fr2/coke	<b>0.200</b>	0.207	<b>0.200</b>	<b>0.200</b>	0.208	0.213	0.205
fr2/desk	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>
fr2/dishes	<b>0.725</b>	1.521	0.837	0.809	1.845	0.731	1.126
fr2/flowerbouquet	<b>0.118</b>	0.139	0.138	0.132	0.137	0.146	0.131
fr2/flowerbouquet_br.	0.652	0.669	0.669	0.655	0.641	<b>0.636</b>	0.703
fr2/large_no_loop	0.623	0.666	0.826	<b>0.522</b>	0.744	0.530	0.887
fr2/large_with_loop	<b>0.371</b>	0.384	0.391	0.380	0.380	0.397	3.598
fr2/metallic_sphere	1.581	1.836	1.586	1.810	0.602	<b>0.546</b>	1.099
fr2/metallic_sphere2	0.821	<b>0.702</b>	0.712	0.774	0.762	0.720	0.772
fr2/pioneer_360	0.213	<b>0.210</b>	<b>0.210</b>	0.215	<b>0.210</b>	0.211	0.213
fr2/pioneer_slam	0.341	0.339	<b>0.332</b>	0.343	0.352	0.343	0.367
fr2/pioneer_slam2	0.411	0.393	0.392	0.391	0.386	0.411	<b>0.381</b>
fr2/pioneer_slam3	0.421	0.393	0.485	0.369	0.514	<b>0.360</b>	0.511
Ortalama	0.513	0.586	0.534	0.523	0.541	<b>0.421</b>	0.762

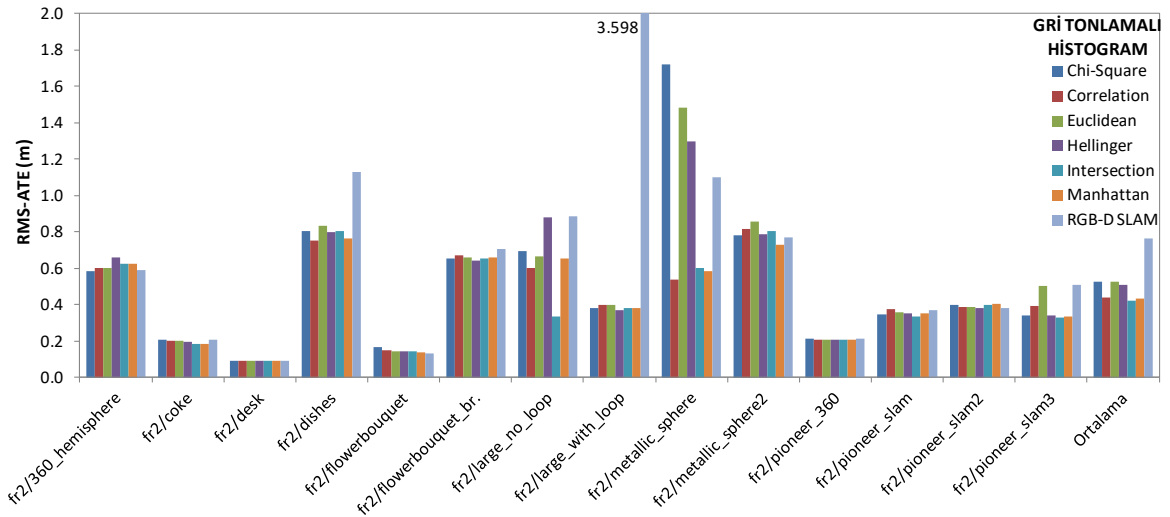
Çizelge 6.3. Gri tonlamalı histogram benzerliği ve sıradüzensel kümeleme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m)

Uzaklık Metriği	Chi-Square	Correlation	Euclidean	Hellinger	Intersection	Manhattan	RGB-D SLAM
fr2/360_hemisphere	<b>0.585</b>	0.600	0.599	0.657	0.621	0.625	0.592
fr2/coke	0.205	0.202	0.202	0.194	<b>0.183</b>	0.184	0.205
fr2/desk	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>
fr2/dishes	0.801	<b>0.749</b>	0.832	0.799	0.806	0.761	1.126
fr2/flowerbouquet	0.168	0.146	0.141	0.145	0.141	0.135	<b>0.131</b>
fr2/flowerbouquet_br.	0.653	0.672	0.661	<b>0.642</b>	0.654	0.660	0.703
fr2/large_no_loop	0.694	0.598	0.664	0.879	<b>0.332</b>	0.653	0.887
fr2/large_with_loop	0.378	0.401	0.399	<b>0.369</b>	0.381	0.379	3.598
fr2/metallic_sphere	1.724	<b>0.539</b>	1.486	1.298	0.599	0.585	1.099
fr2/metallic_sphere2	0.780	0.813	0.856	0.789	0.802	<b>0.730</b>	0.772
fr2/pioneer_360	0.209	<b>0.204</b>	<b>0.204</b>	0.205	0.205	0.205	0.213
fr2/pioneer_slam	0.343	0.377	0.355	0.349	<b>0.333</b>	0.349	0.367
fr2/pioneer_slam2	0.398	0.387	0.385	<b>0.379</b>	0.400	0.401	0.381
fr2/pioneer_slam3	0.341	0.389	0.501	0.337	<b>0.331</b>	0.335	0.511
Ortalama	0.526	0.441	0.527	0.509	<b>0.420</b>	0.435	0.762

Genel ortalamaya bakıldığında en iyi sonuçlar Gri+Intersection ve RGB+Manhattan kombinasyonları ile sırasıyla 42 cm ve 42.1 cm şeklinde elde edilmiştir. RGB ya da gri tonlamalı histogram kullanımı, Correlation ve Intersection metrikleri dışında ortalamada birbirine yakın sonuçlar üretmektedir. Bu iki metrik için gri tonlamalı histogram kullanıldığında sırasıyla yaklaşık 15 cm ve 12 cm daha düşük hata gözlenmiştir. Histogram karşılaştırmasından sonra sıradüzensel kümeleme tabanlı nitelik eşleştirmesi uygulanması, *brute-force* eşleştirme tekniğine benzer şekilde kayma hatasının önemli derecede düşmesini sağlamıştır. Çizelge 5.7 ve Çizelge 5.8 ile birlikte değerlendirme yapıldığında, sıradüzensel kümeleme ve *brute-force* yaklaşımların doğruluk açısından çok yakın sonuçlar ürettikleri gözlenmiştir. Bu durum, her bir anahtar çerçeve için yalnızca bir kümeleme ağacı kullanılmasının eşleşme aramasında yeterli keskinliği sağladığını göstermektedir. Bu etki Şekil 6.6 ve Şekil 6.7’de görülebilmektedir. Bu teknik çerçevesinde Gri+Intersection kombinasyonu, ortalama hatanın RGB-D SLAM sistemine [44] göre yaklaşık %45 oranında azaltılmasını sağlamıştır.



Şekil 6.6. RGB histogram benzerliği ve sıradüzensel kümeleme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması



Şekil 6.7. Gri tonlamalı histogram benzerliği ve sıradüzensel kümeleme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması

### 6.3.2. Autocorrelogram Benzerliği ile Sıradüzensel Kümeleme Deneyleri

Bu kesimde ise sıradüzensel kümeleme yöntemiyle nitelik eşleştirmesi, autocorrelogram karşılaştırmasına göre belirlenen  $4 \times 8 = 32$  anahtar çerçevesel başlangıç grubu üzerinde gerçekleştirilerek mevcut çerçeve ile eşleşme sayısına göre 8 döngü kapama adayı belirlenmiştir. Çizelge 6.4 ve Çizelge 6.5, sırasıyla RGB ve gri tonlamalı autocorrelogram kullanımıyla elde edilen RMS-ATE sonuçlarını içermektedir.

Çizelge 6.4. RGB autocorrelogram benzerliği ve sıradüzensel kümeleme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m)

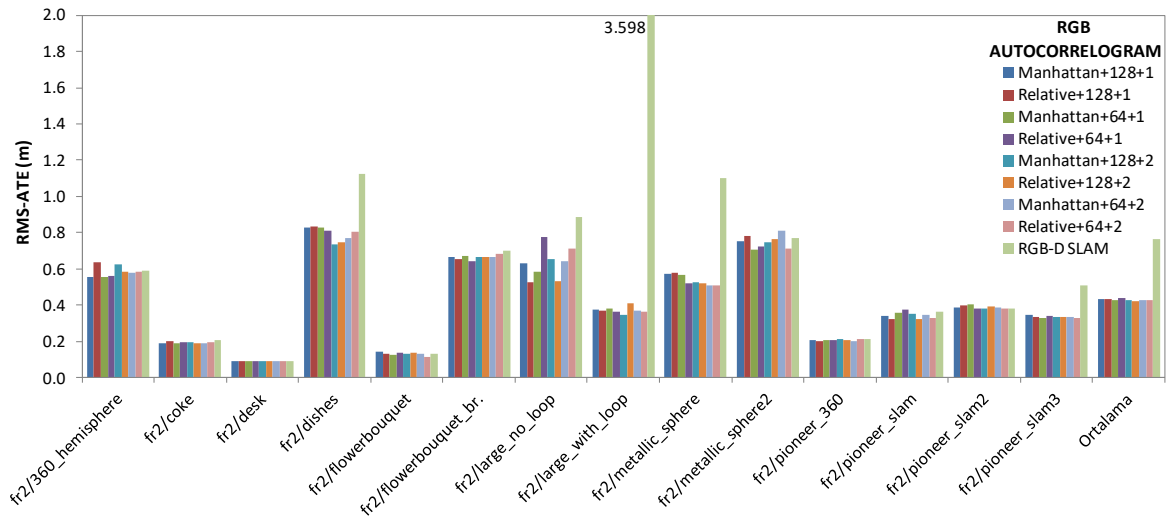
Nicelendirme Faktörü	64				128				RGB-D SLAM
	1		2		1		2		
Piksel Uzaklığı	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	
Uzaklık Metriği	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	
fr2/360_hemisphere	0.556	0.561	0.578	0.582	<b>0.554</b>	0.638	0.623	0.582	0.592
fr2/coke	<b>0.187</b>	0.196	0.192	0.197	0.193	0.199	0.198	0.189	0.205
fr2/desk	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>
fr2/dishes	0.829	0.808	0.770	0.806	0.828	0.832	<b>0.735</b>	0.747	1.126
fr2/flowerbouquet	0.123	0.139	0.131	<b>0.112</b>	0.142	0.135	0.130	0.136	0.131
fr2/flowerbouquet_br.	0.670	<b>0.643</b>	0.664	0.681	0.665	0.653	0.666	0.665	0.703
fr2/large_no_loop	0.583	0.778	0.642	0.711	0.632	<b>0.527</b>	0.652	0.534	0.887
fr2/large_with_loop	0.383	0.364	0.372	0.365	0.377	0.368	<b>0.349</b>	0.409	3.598
fr2/metallic_sphere	0.569	0.523	<b>0.508</b>	0.509	0.573	0.578	0.528	0.518	1.099
fr2/metallic_sphere2	<b>0.705</b>	0.724	0.812	0.714	0.752	0.782	0.747	0.767	0.772
fr2/pioneer_360	0.206	0.210	<b>0.200</b>	0.211	0.206	0.202	0.211	0.210	0.213
fr2/pioneer_slam	0.358	0.374	0.344	0.328	0.340	<b>0.324</b>	0.353	<b>0.324</b>	0.367
fr2/pioneer_slam2	0.403	0.381	0.389	<b>0.380</b>	0.389	0.401	0.383	0.392	0.381
fr2/pioneer_slam3	<b>0.328</b>	0.344	0.337	0.332	0.345	0.336	0.334	0.333	0.511
Ortalama	0.428	0.438	0.431	0.430	0.435	0.433	0.429	<b>0.421</b>	0.762

Çizelge 6.5. Gri tonlamalı autocorrelogram benzerliği ve sıradüzensel kümeleme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m)

Nicelendirme Faktörü	64				128				RGB-D SLAM
	1		2		1		2		
Piksel Uzaklığı	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	
Uzaklık Metriği	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	
fr2/360_hemisphere	0.666	<b>0.520</b>	0.616	0.564	0.721	0.575	0.656	0.538	0.592
fr2/coke	0.206	0.201	0.206	0.191	0.184	0.196	0.198	<b>0.183</b>	0.205
fr2/desk	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>
fr2/dishes	0.785	0.788	0.815	<b>0.739</b>	0.761	0.754	0.834	0.774	1.126
fr2/flowerbouquet	0.139	0.132	0.160	0.157	0.149	0.143	<b>0.130</b>	0.142	0.131
fr2/flowerbouquet_br.	0.656	0.676	0.672	0.685	0.654	0.643	<b>0.635</b>	0.651	0.703
fr2/large_no_loop	0.559	0.672	0.724	0.519	0.746	0.711	0.674	<b>0.409</b>	0.887
fr2/large_with_loop	<b>0.357</b>	0.377	0.370	0.377	0.381	0.370	0.386	0.362	3.598
fr2/metallic_sphere	0.551	0.643	0.539	0.820	0.507	<b>0.502</b>	0.558	0.537	1.099
fr2/metallic_sphere2	0.717	0.758	0.822	<b>0.710</b>	0.759	0.765	0.817	0.766	0.772
fr2/pioneer_360	0.214	0.209	<b>0.206</b>	0.212	0.208	0.209	0.213	0.210	0.213
fr2/pioneer_slam	<b>0.335</b>	0.339	0.376	0.341	0.350	0.345	0.345	0.355	0.367
fr2/pioneer_slam2	0.395	0.396	0.400	0.392	0.385	0.387	0.391	0.401	<b>0.381</b>
fr2/pioneer_slam3	0.333	<b>0.327</b>	0.339	0.337	0.334	0.339	0.338	0.333	0.511
Ortalama	0.429	0.438	0.453	0.438	0.445	0.431	0.447	<b>0.411</b>	0.762

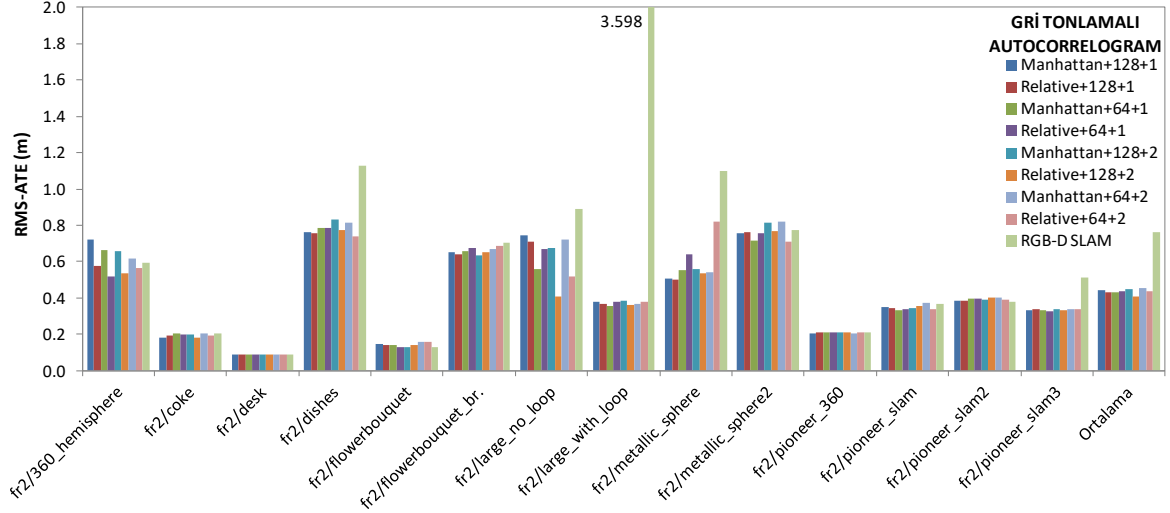


Sonuçlar incelendiğinde, sıradüzensel kümeleme tabanlı nitelik eşleştirmesi yoluyla yerel görüntü niteliklerinin kullanılmasının, sadece autocorrelogram benzerliğine göre karar verilmesinden (Bkz. Çizelge 5.5 ve Çizelge 5.6) daha başarılı sonuçlar ürettiği görülmektedir. Ortalamada tüm kombinasyonlar için bu teknikle daha düşük RMS-ATE değerleri elde edilmiştir. Bir önceki kesimdeki sonuçlara benzer şekilde sıradüzensel kümeleme tekniğinin, kaynak çerçevedeki her bir anahtar nokta için hedef çerçevedeki tüm anahtar noktalar ile mesafe hesaplayan *brute-force* yaklaşıma çok yakın performans sergilediği görülmektedir (Bkz. Çizelge 5.9 ve Çizelge 5.10). Dolayısıyla, anahtar çerçeve başına tek bir kümeleme ağacı oluşturulmasının yeterli derecede başarılı nitelik eşleştirmesi sağladığı anlaşılmaktadır.



Şekil 6.8. RGB autocorrelogram benzerliği ve sıradüzensel kümeleme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması

*Brute-force* yönteme benzer şekilde burada da farklı parametre kombinasyonları ile elde edilen hata miktarları birbirine son derece yakın olup 41.1 cm - 45.3 cm aralığındadır. Bu durumu sağlayan temel etken, nicelendirme faktörü ve piksel mesafesi parametrelerinin sonuçlar üzerinde çok düşük etkiye sahip olmasıdır. Şekil 6.8 ve Şekil 6.9'dan da anlaşılacağı üzere, RGB ya da gri tonlamalı autocorrelogram kullanımı birbirine yakın sonuçlar üretmektedir.



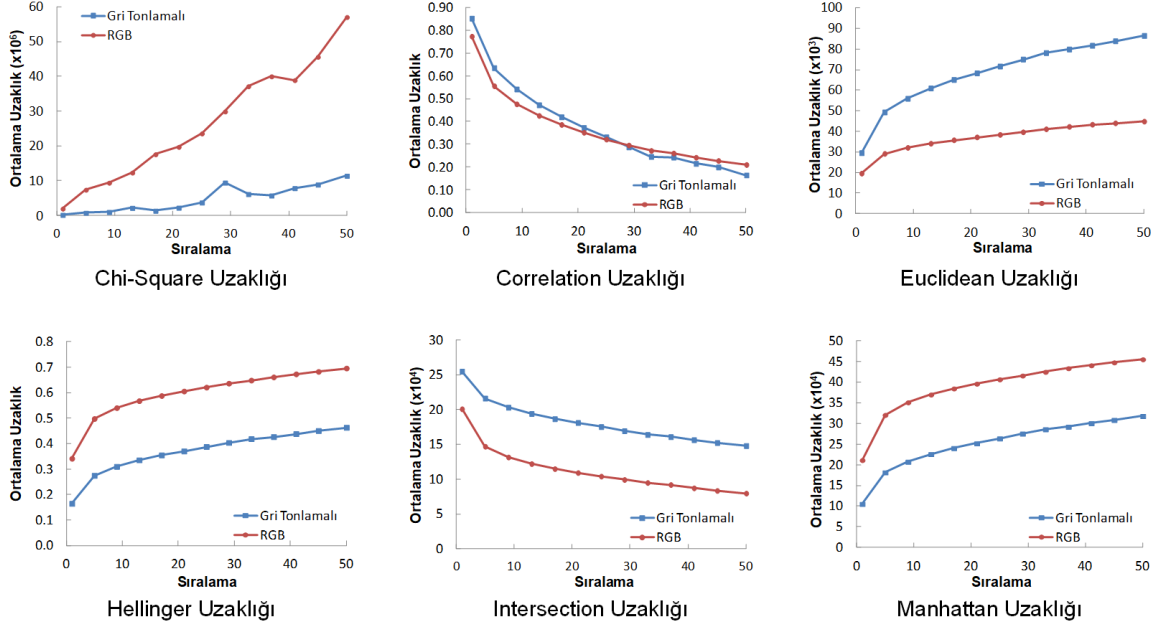
Şekil 6.9. Gri tonlamalı autocorrelogram benzerliği ve sıradüzensel kümeleme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması

Bu teknikle elde edilen en başarılı ortalama sonuç, Gri+Relative+128+2 kombinasyonunun ürettiği 41.1 cm'lik kayma hatasıdır. Bu sonuç, RGB-D SLAM sistemine [44] göre yaklaşık %46 düzeyinde doğruluğun artırılması anlamına gelmektedir. Bu oran, tüm deneyler boyunca elde edilen en yüksek iyileştirme sonucu, dolayısıyla gözlemlenen en düşük mutlak yol hatasıdır.

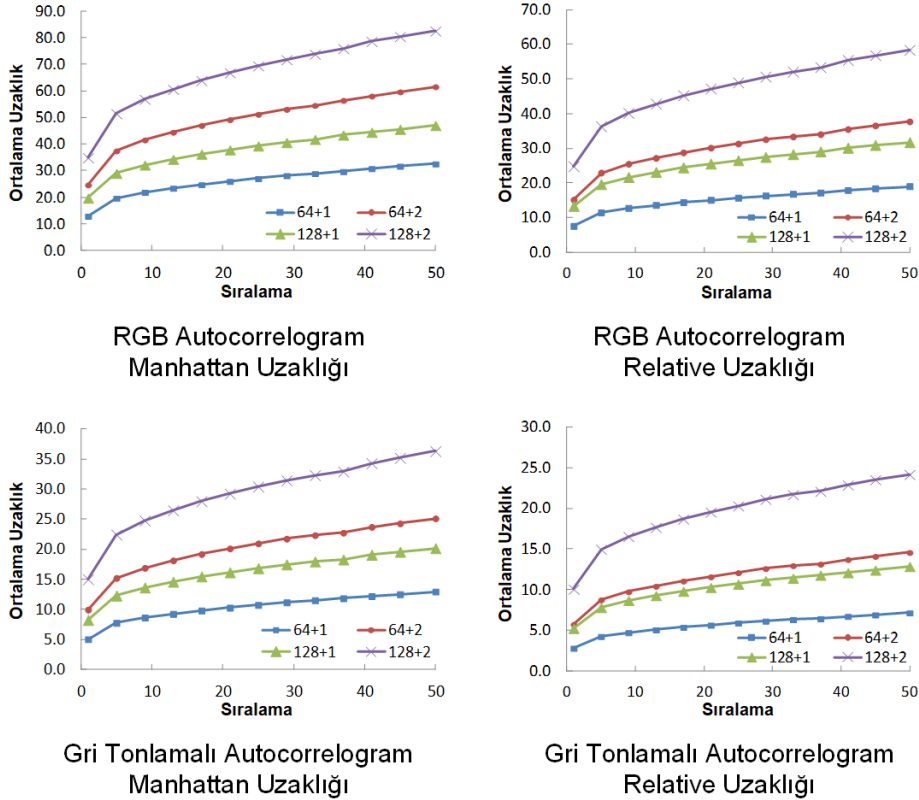
### 6.3.3. Uyarlanabilir Eşikleme için Benzerlik Eşiğinin Belirlenmesi

Uyarlanabilir eşikleme tekniğinin uygulanabilmesi için öncelikle benzerlik eşiğinin ( $gruptaki\_en\_iyi\_sonuç \times eşikleme\_faktörü$ ) belirlenmesi gereklidir. Bu kapsamda öncelikle,  $eşikleme\_faktörü$  parametresinin değerine karar verilebilmesi için bir analiz gerçekleştirilmiştir. Bu amaçla, tüm görüntü kayıtları üzerinde her bir uzaklık metriği kullanılarak elde edilen genel tanımlayıcı benzerlik sonuçları incelenmiştir.

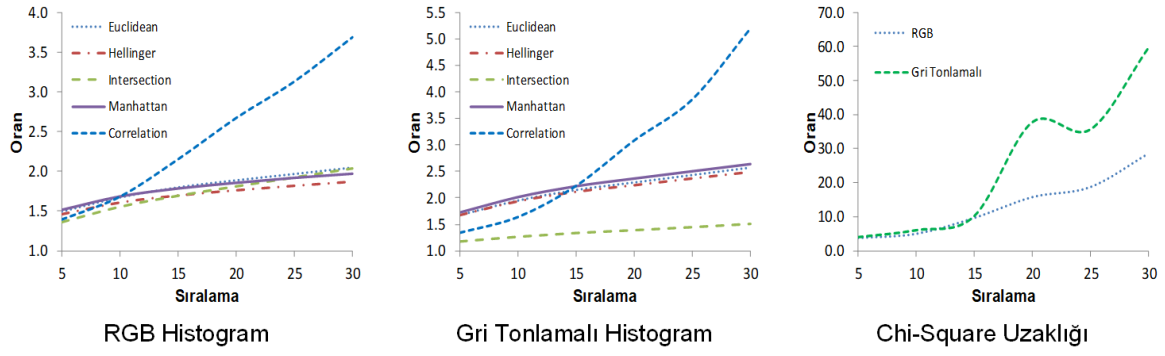
Şekil 6.10 ve Şekil 6.11, sırasıyla histogram ve autocorrelogram kullanımıyla, her bir uzaklık metriği için aday sırasına göre ortalama uzaklık değerlerinin değişimini göstermektedir. Şekillerde görüldüğü üzere mevcut çerçeveye benzerlik oranı, en benzer aday ile beşinci sıradaki aday arasında hızlı bir düşüş göstermekte, daha sonra ise yavaş şekilde azalmaktadır. Buradan hareketle, mevcut çerçeveye en benzer (birinci sıradaki) aday ile diğer adaylar arasındaki benzerlik sonuçlarının oranları hesaplanmıştır.



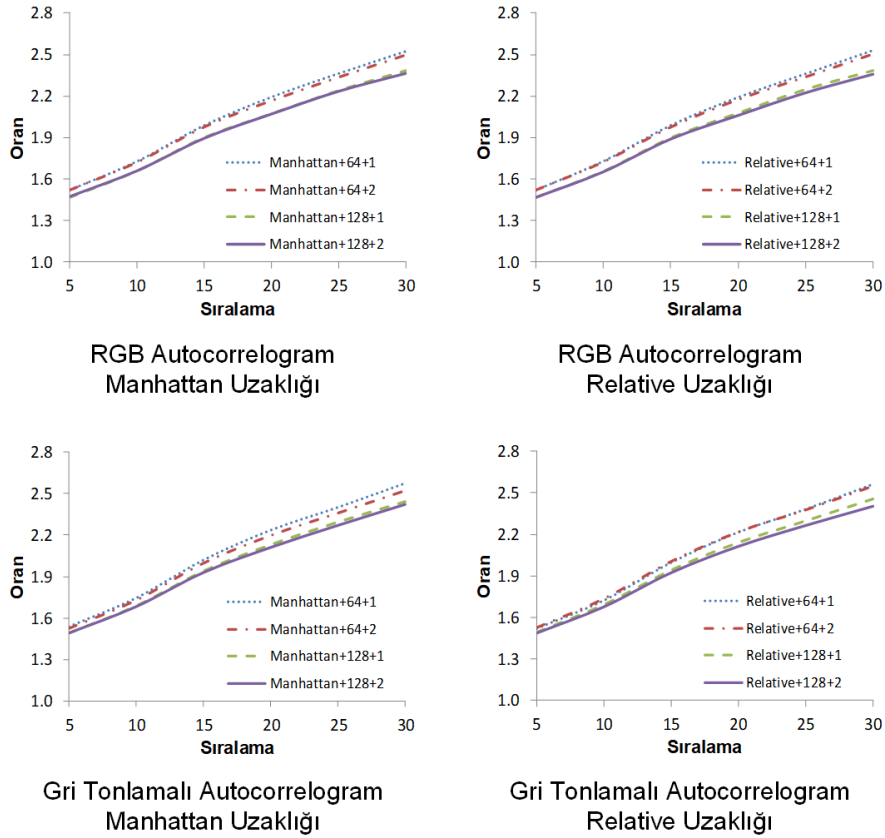
Şekil 6.10. Aday sıralamasına göre, tüm görüntü kayıtları üzerinde ortalama histogram benzerliği dağılımı (Diğer metriklerden farklı olarak, Correlation ve Intersection için yüksek mesafe değeri daha fazla benzerlik anlamına gelmektedir)



Şekil 6.11. Aday sıralamasına göre, tüm görüntü kayıtları üzerinde ortalama autocorrelogram benzerliği dağılımı



Şekil 6.12. En benzer aday ile diğer adaylar arasındaki histogram benzerliği oranının değişimi (Chi-Square uzaklığı yüksek değerler ürettiğinden farklı bir grafikte gösterilmiştir)



Şekil 6.13. En benzer aday ile diğer adaylar arasındaki autocorrelogram benzerliği oranının değişimi

Şekil 6.12'de görülen histogram benzerlik oranları, Chi-Square ve Correlation dışındaki metriklerde benzer sonuçlar elde edildiğini göstermektedir. Özellikle Chi-Square metriği kullanımında aday çerçevelerin mevcut çerçeveye olan

benzerlikleri çok hızlı şekilde azalmaktadır. Intersection metriği ise gri tonlamalı histogram ile birlikte kullanıldığında, farklı karakteristiğinden dolayı uzaklık daha yavaş şekilde değişmektedir. Şekil 6.13'teki veriler ise, autocorrelogram benzerlik oranlarının tüm metrikler ve parametre kombinasyonları için benzer şekilde değiştiğini göstermektedir.

Çizelge 6.6. Histogram kullanımında *eşikleme\_faktörü* parametresi için belirlenen değerler

Uzaklık Metriği	RGB	Gri Tonlamalı
Chi-Square	6	7
Correlation	2.7	3
Euclidean	2	2.5
Hellinger	2	2.5
Intersection	2	1.5
Manhattan	2	2.5

Çizelge 6.7. Autocorrelogram kullanımında *eşikleme\_faktörü* parametresi için belirlenen değerler

Uzaklık Metriği	Nicelendirme Faktörü	
	64	128
Manhattan	2.2	2.1
Relative	2.2	2.1

Gerçekleştirilen analizlerden hareketle, döngü kapama adaylarının belirlenmesi aşamasında başlangıç aday grubunun  $4 \times 8 = 32$  anahtar çerçeveden oluştuğu hesaba katılarak, *eşikleme\_faktörü* parametresi için 1. ve 30. sıradaki adaylar arasındaki benzerlik oranı kullanılmıştır. Dolayısıyla, Şekil 6.12 ve Şekil 6.13'teki veriler kullanılarak, *eşikleme\_faktörü* için Çizelge 6.6 ve Çizelge 6.7'de yer alan değerler uygulanmıştır. Dolayısıyla örneğin RGB histogram ve Euclidean metriği birlikte kullanıldığında, herhangi bir döngü kapama aramasında başlangıç grubu içerisinde mevcut çerçeveye *gruptaki\_en\_ iyi\_sonuç*  $\times 2$ 'den daha az benzer olan adaylar elenmektedir. Burada istisna olarak Correlation ve Intersection metriklerinde benzerlik arttıkça skor da arttığından, bu metrikler için *eşikleme\_faktörü* çarpmaya göre tersi alınarak uygulanmıştır. Benzer şekilde

autocorrelogram için, diğer parametrelere bakılmaksızın nicelendirme faktörü olarak 128 kullanıldığında *gruptaki\_en\_ iyi\_sonuç* × 2.1'den, 64 kullanıldığında ise *gruptaki\_en\_ iyi\_sonuç* × 2.2'den daha düşük benzerliğe sahip adaylar başlangıç grubundan elenmektedir.

Bu analiz çalışması neticesinde uyarlanabilir eşikleme için *eşikleme\_faktörü* değerleri belirlenmiş olup, sonraki deneylerde modele aykırı adayların her bir döngü kapama araması için dinamik şekilde elenmesi amacıyla kullanılmışlardır.

#### 6.3.4. Histogram Benzerliği ile Uyarlanabilir Eşikleme Deneyleri

Bu kesimde, uyarlanabilir eşikleme bileşeninin de aktif hale getirilmesiyle, önerilen genişletilmiş döngü kapama tespiti yöntemi Şekil 6.1'de görülen tüm bileşenlerinin dâhil olduğu komple haliyle test edilmiştir. İlk aşamada histogram benzerliğine göre seçilen  $4 \times 8 = 32$  anahtar çerçeveden oluşan başlangıç grubu üzerinde uyarlanabilir eşikleme uygulanarak modele aykırı adaylar elenmektedir. Sonrasında ise eksik kümeleme ağaçları oluşturulmakta ve her bir anahtar çerçeve ile mevcut çerçeve arasında sıradüzensel kümeleme tabanlı nitelik eşleştirmesi gerçekleştirilmektedir. Son adımda ise mevcut çerçeve ile en fazla eşleşme üreten 8 anahtar çerçeve döngü kapama adayı olarak seçilmektedir. RGB ve gri tonlamalı histogram kullanımı için mutlak yol hatası sonuçları sırasıyla Çizelge 6.8 ve Çizelge 6.9'da sunulmaktadır.

Çizelge 6.8. RGB histogram benzerliği ve uyarlanabilir eşikleme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m)

Uzaklık Metriği	Chi-Square	Correlation	Euclidean	Hellinger	Intersection	Manhattan	RGB-D SLAM
fr2/360_hemisphere	0.614	0.662	0.746	0.623	<b>0.550</b>	0.627	0.592
fr2/coke	0.197	<b>0.184</b>	0.195	0.192	0.190	0.213	0.205
fr2/desk	<b>0.089</b>	<b>0.089</b>	<b>0.089</b>	<b>0.089</b>	0.090	<b>0.089</b>	0.090
fr2/dishes	<b>0.763</b>	0.824	0.829	0.781	0.807	0.765	1.126
fr2/flowerbouquet	0.125	<b>0.112</b>	0.151	0.142	0.131	0.133	0.131
fr2/flowerbouquet_br.	0.731	0.754	0.779	<b>0.680</b>	0.728	0.762	0.703
fr2/large_no_loop	0.719	0.799	0.919	0.501	0.724	<b>0.348</b>	0.887
fr2/large_with_loop	0.392	0.375	0.409	0.389	0.375	<b>0.364</b>	3.598
fr2/metallic_sphere	1.402	1.713	1.392	1.662	<b>0.765</b>	0.967	1.099
fr2/metallic_sphere2	0.780	<b>0.645</b>	0.765	0.810	0.696	0.716	0.772
fr2/pioneer_360	0.210	0.213	0.232	0.210	<b>0.202</b>	0.212	0.213
fr2/pioneer_slam	0.321	0.338	<b>0.319</b>	0.344	0.344	0.378	0.367
fr2/pioneer_slam2	0.395	0.407	<b>0.368</b>	0.388	0.386	0.374	0.381
fr2/pioneer_slam3	0.391	0.743	0.515	<b>0.346</b>	0.355	0.460	0.511
Ortalama	0.509	0.561	0.551	0.511	<b>0.453</b>	0.458	0.762

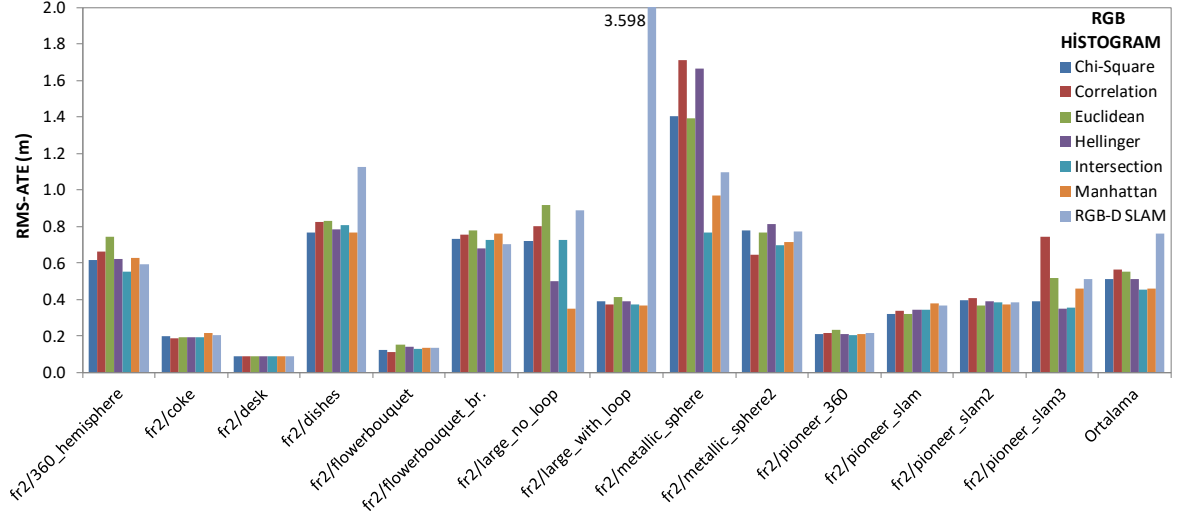
Çizelge 6.9. Gri tonlamalı histogram benzerliği ve uyarlanabilir eşikleme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m)

Uzaklık Metriği	Chi-Square	Correlation	Euclidean	Hellinger	Intersection	Manhattan	RGB-D SLAM
fr2/360_hemisphere	0.632	0.614	0.682	0.601	<b>0.569</b>	0.577	0.592
fr2/coke	0.200	0.207	0.206	<b>0.185</b>	0.194	0.198	0.205
fr2/desk	0.090	0.089	<b>0.088</b>	0.089	0.090	0.089	0.090
fr2/dishes	0.774	0.780	0.808	<b>0.718</b>	0.797	0.880	1.126
fr2/flowerbouquet	0.166	0.149	0.140	0.137	0.137	0.144	<b>0.131</b>
fr2/flowerbouquet_br.	0.766	0.723	0.783	0.730	0.723	0.733	<b>0.703</b>
fr2/large_no_loop	0.859	0.648	0.604	0.510	<b>0.395</b>	0.430	0.887
fr2/large_with_loop	0.388	0.381	0.385	<b>0.348</b>	0.367	0.381	3.598
fr2/metallic_sphere	1.576	1.338	1.614	1.661	0.914	<b>0.512</b>	1.099
fr2/metallic_sphere2	0.782	0.739	0.755	0.859	<b>0.688</b>	0.781	0.772
fr2/pioneer_360	<b>0.205</b>	0.207	0.215	0.216	0.213	0.207	0.213
fr2/pioneer_slam	<b>0.330</b>	0.373	0.347	0.357	0.349	0.345	0.367
fr2/pioneer_slam2	0.394	0.416	0.391	0.392	0.400	0.390	<b>0.381</b>
fr2/pioneer_slam3	<b>0.322</b>	0.477	0.521	0.357	0.341	0.527	0.511
Ortalama	0.535	0.510	0.539	0.511	<b>0.441</b>	0.443	0.762

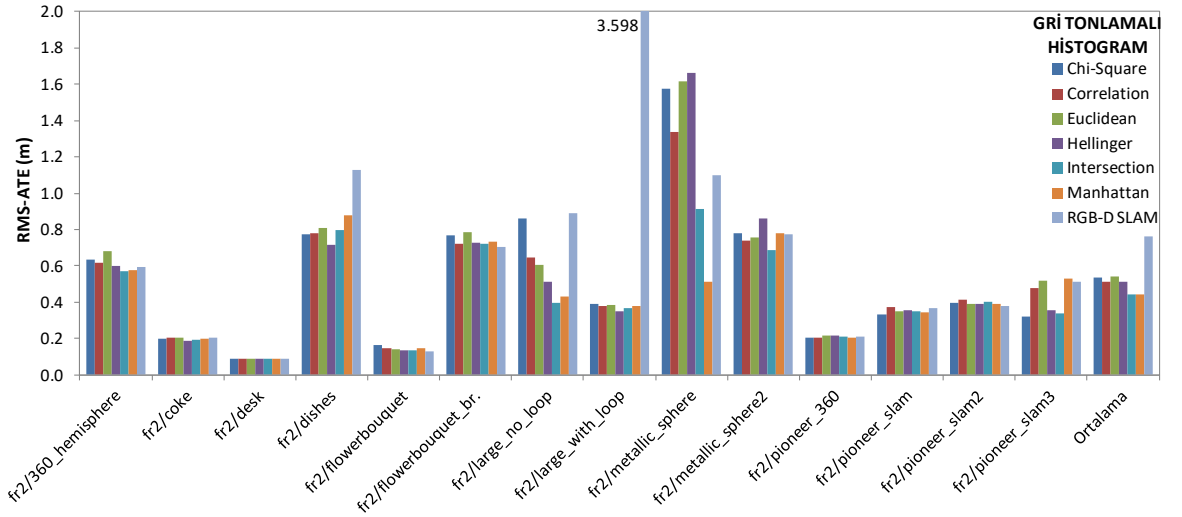
Elde edilen sonuçlara göre, Intersection ve Manhattan metriklerinin sırasıyla gri tonlamalı ve RGB histogram kullanımıyla en başarılı sonuçları ürettikleri görülmektedir. Ortalamada en iyi sonuç 44.1 cm RMS-ATE ile Gri+Intersection ile elde edilmiş olup, 44.3 cm hata ile çok yakın bir performans sergileyen Gri+Manhattan kombinasyonu ikinci sıradadır. RGB histogram kullanıldığında ise yine bu iki metrik sırasıyla 45.3 cm ve 45.8 cm hata oranlarıyla en başarılıya yakın sonuçlar üretmişlerdir.

Yerel niteliklerin histogram benzerliği ile birlikte kullanıldığı diğer deney sonuçlarına benzer olarak, gri tonlamalı ya da RGB histogram kullanımı burada da yakın performans sergilemektedir (Bkz. Şekil 6.14 ve Şekil 6.15). Bu durum, her iki histogram türüyle de seçilen başlangıç grubunun yeterince kapsayıcı olduğunu göstermektedir. Çizelge 5.3 ve Çizelge 5.4'teki sonuçlar hesaba katılarak yalnızca histogram benzerliği kullanılan teknikle karşılaştırma yapıldığında, uyarlanabilir eşiklemenin sıradüzensel kümeleme ile birlikte kullanıldığı bu teknik ortalama gri tonlamalı kombinasyonlar için yaklaşık %26 (~18 cm), RGB kombinasyonlar için ise yaklaşık %18 (~11 cm) oranında hatanın ekstra azaltılmasını sağlamaktadır. Sıradüzensel kümeleme tekniği ile karşılaştırıldığında ise (Bkz. Çizelge 6.2 ve Çizelge 6.3) yakın sonuçlar elde edildiği görülmüştür. Ayrıca bu teknik RGB-D

SLAM'in [44] doğruluğunu yaklaşık %42 düzeyinde (Gri+Intersection kombinasyonu ile) arttırmaktadır.



Şekil 6.14. RGB histogram benzerliği ve uyarlanabilir eşikleme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması



Şekil 6.15. Gri tonlamalı histogram benzerliği ve uyarlanabilir eşikleme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması



### 6.3.5. Autocorrelogram Benzerliği ile Uyarlanabilir Eşikleme Deneyleri

Bu deney grubunda ise, genişletilmiş döngü kapama tespiti yöntemi bir önceki kesimdeki gibi tüm bileşenleriyle test edilmiş ve genel tanımlayıcı olarak autocorrelogram kullanılmıştır.  $4 \times 8 = 32$  anahtar çerçevelik başlangıç grubu autocorrelogram benzerliğine göre belirlenmiş, sonrasında ise Bölüm 6.3.3'te hesaplanan benzerlik eşik değeri kullanılarak uyarlanabilir eşikleme gerçekleştirilmiş ve daha az benzer adaylar dinamik şekilde elenmiştir. Kümeleme ağacı oluşturma aşaması da geçildikten sonra sıradüzensel kümeleme uygulanarak anahtar nokta eşleştirmesi gerçekleştirilerek eşleşme sayılarına göre 8 döngü kapama adayı seçilmiştir. Çizelge 6.10 ve Çizelge 6.11 sırasıyla RGB ve gri tonlamalı autocorrelogram için bu tekniğe dair kayma hatası sonuçlarını içermektedir.

Çizelge 6.10. RGB autocorrelogram benzerliği ve uyarlanabilir eşikleme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m)

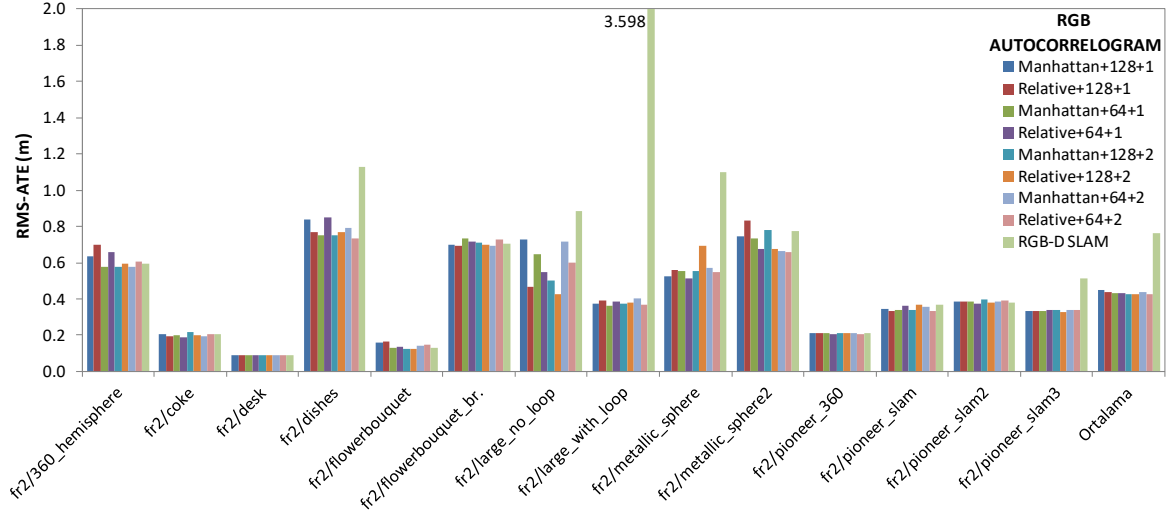
Nicelendirme Faktörü	64				128				RGB-D SLAM
	1		2		1		2		
Piksel Uzaklığı	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	
fr2/360_hemisphere	<b>0.579</b>	0.661	0.580	0.606	0.637	0.700	<b>0.579</b>	0.593	0.592
fr2/coke	0.200	<b>0.188</b>	0.193	0.207	0.207	0.195	0.215	0.198	0.205
fr2/desk	<b>0.089</b>	0.090	0.090	0.090	0.090	0.090	0.091	0.090	0.090
fr2/dishes	0.751	0.852	0.793	<b>0.731</b>	0.838	0.770	0.751	0.767	1.126
fr2/flowerbouquet	0.132	0.134	0.141	0.145	0.157	0.164	<b>0.123</b>	0.126	0.131
fr2/flowerbouquet_br.	0.733	0.719	<b>0.692</b>	0.726	0.702	0.696	0.709	0.699	0.703
fr2/large_no_loop	0.649	0.546	0.718	0.599	0.728	0.465	0.504	<b>0.425</b>	0.887
fr2/large_with_loop	<b>0.362</b>	0.383	0.400	0.369	0.374	0.390	0.376	0.380	3.598
fr2/metallic_sphere	0.552	<b>0.513</b>	0.569	0.550	0.525	0.557	0.552	0.693	1.099
fr2/metallic_sphere2	0.735	0.675	0.662	<b>0.658</b>	0.744	0.835	0.780	0.674	0.772
fr2/pioneer_360	0.212	<b>0.205</b>	0.213	0.208	0.209	0.209	0.212	0.212	0.213
fr2/pioneer_slam	0.338	0.365	0.359	0.335	0.346	<b>0.333</b>	0.340	0.365	0.367
fr2/pioneer_slam2	0.388	<b>0.375</b>	0.386	0.391	0.384	0.388	0.396	0.382	0.381
fr2/pioneer_slam3	0.331	0.341	0.337	0.340	0.332	0.336	0.338	<b>0.327</b>	0.511
Ortalama	0.432	0.432	0.438	0.425	0.448	0.438	0.426	<b>0.424</b>	0.762

Çizelge 6.11. Gri tonlamalı autocorrelogram benzerliği ve uyarlanabilir eşikleme tekniği için mutlak yol hatası (RMS-ATE) sonuçları (m)

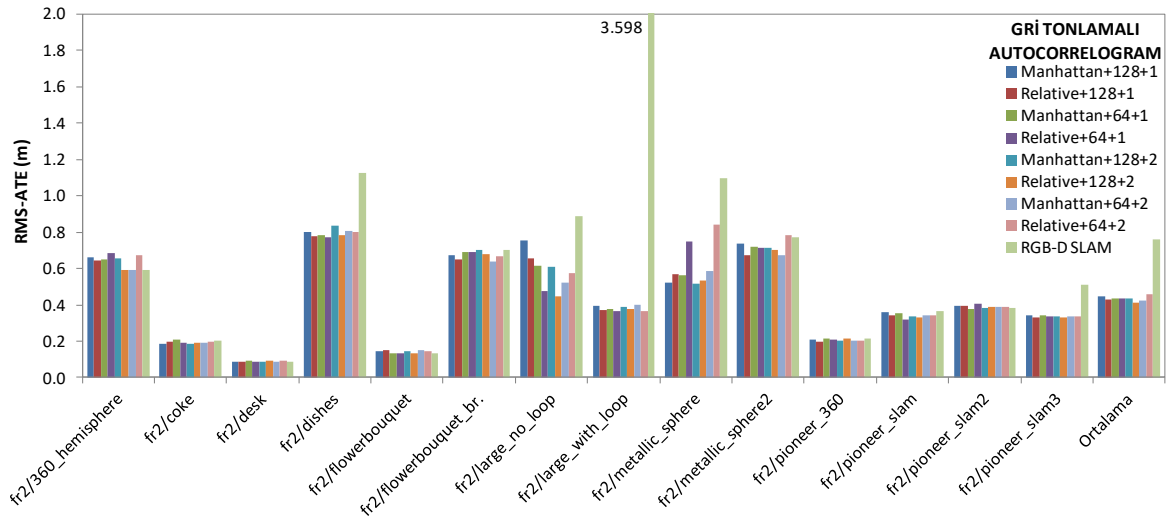
Nicelendirme Faktörü	64				128				RGB-D SLAM
	1		2		1		2		
Piksel Uzaklığı	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	Manhattan	Relative	
fr2/360_hemisphere	0.652	0.686	0.593	0.676	0.664	0.645	0.654	0.595	<b>0.592</b>
fr2/coke	0.209	0.190	0.189	0.196	<b>0.183</b>	0.197	0.188	0.191	0.205
fr2/desk	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>	<b>0.090</b>
fr2/dishes	0.782	<b>0.773</b>	0.809	0.803	0.801	0.777	0.835	0.785	1.126
fr2/flowerbouquet	<b>0.131</b>	<b>0.131</b>	0.151	0.145	0.146	0.152	0.144	0.132	<b>0.131</b>
fr2/flowerbouquet_br.	0.689	0.690	<b>0.641</b>	0.670	0.675	0.649	0.702	0.678	0.703
fr2/large_no_loop	0.616	0.476	0.523	0.572	0.756	0.654	0.611	<b>0.449</b>	0.887
fr2/large_with_loop	0.377	0.368	0.401	<b>0.363</b>	0.395	0.373	0.387	0.377	3.598
fr2/metallic_sphere	0.564	0.747	0.585	0.843	0.525	0.569	<b>0.517</b>	0.535	1.099
fr2/metallic_sphere2	0.722	0.712	<b>0.671</b>	0.783	0.739	<b>0.671</b>	0.716	0.703	0.772
fr2/pioneer_360	0.215	0.210	0.205	0.205	0.206	<b>0.200</b>	0.201	0.213	0.213
fr2/pioneer_slam	0.354	<b>0.318</b>	0.344	0.343	0.360	0.342	0.338	0.331	0.367
fr2/pioneer_slam2	<b>0.377</b>	0.404	0.390	0.391	0.396	0.394	0.381	0.392	0.381
fr2/pioneer_slam3	0.340	0.334	0.337	0.338	0.340	0.333	0.334	<b>0.330</b>	0.511
Ortalama	0.437	0.438	0.423	0.459	0.448	0.432	0.436	<b>0.414</b>	0.762

Elde edilen en düşük ortalama hata, Gri+Relative+128+2 kombinasyonu tarafından üretilen 41.4 cm'dir. Genel olarak 2 piksel mesafesine göre oluşturulan autocorrelogram kombinasyonları ortalama sonuçlarda daha üst sıralarda yer alsalar da, 1 piksel mesafesi ile karşılaştırıldığında 2 cm - 3 cm civarında bir değişim gözlenmektedir. Görüntü nicelendirmesinin 128 ya da 64 renk / gri tonu seviyesine göre yapılması da yine ortalamada yaklaşık 1 cm - 2 cm aralığında değişime sebep olmaktadır. Dolayısıyla autocorrelogram kullanılan diğer tekniklerde olduğu gibi bu teknikte de nicelendirme faktörü ve piksel mesafesi parametreleri sonuçlar üzerinde minimal değişime neden olmaktadır (Bkz. Şekil 6.16 ve Şekil 6.17). Benzer şekilde RGB ya da gri tonlamalı autocorrelogram kombinasyonları da yakın performansa sahiptir. Ortalama sonuçlara bakıldığında kayma hatalarının 41.4 cm - 45.9 cm gibi çok dar bir aralıkta kümelendiği görülmektedir. Çizelge 6.4 ve Çizelge 6.5'teki veriler göz önünde bulundurularak, sıradüzensel kümeleme tekniği ile karşılaştırma yapıldığında sonuçlar arasında genellikle çok düşük farklar olduğu gözlenmektedir. Bu durum, uyarlanabilir

eşikleme yaklaşımı uygulandığında yöntemin başarımını koruduğunu göstermektedir.



Şekil 6.16. RGB autocorrelogram benzerliği ve uyarlanabilir eşikleme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması



Şekil 6.17. Gri tonlamalı autocorrelogram benzerliği ve uyarlanabilir eşikleme tekniği ile RGB-D SLAM [44] arasında doğruluk karşılaştırması

Bu teknikle elde edilen en başarılı sonuç olan 41.4 cm, yöntemin entegre edildiği RGB-D SLAM [44] ile kıyaslandığında yaklaşık %46 oranında başarı artışını ifade etmektedir. Bu sonuç, sıradüzensel kümeleme tekniği ile elde edilen en iyi

sonuçtan (Gri+Relative+128+2 kombinasyonunun ürettiği 41.1 cm, Bkz. Çizelge 6.5) 3 mm daha yüksek kayma hatasını gösterse de, bir sonraki kesimde de açıklandığı üzere sağladığı işlemsel performans avantajıyla tüm teknikler içerisindeki en etkili kombinasyondur.

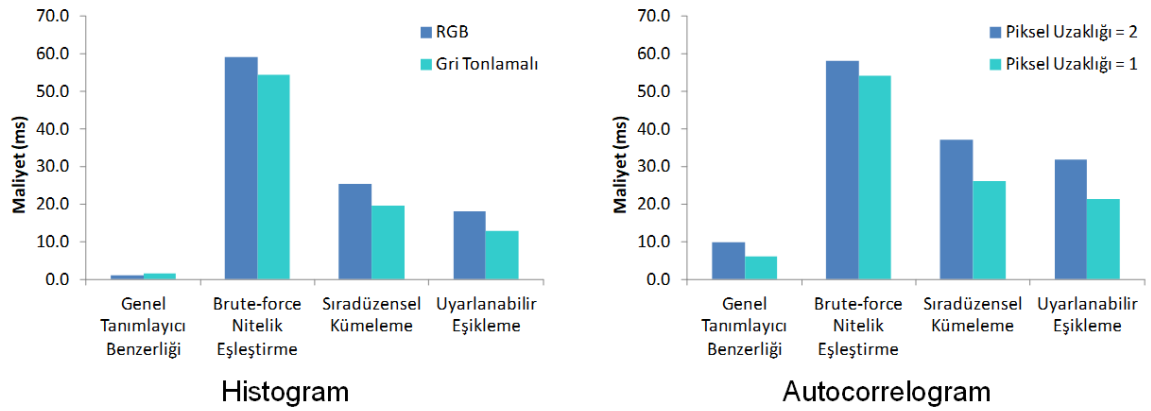
### 6.3.6. İşlemsel Performans ve Tartışma

Çizelge 6.12'de, genişletilmiş döngü kapama tespiti yönteminin yeni bileşenleri ile elde edilen çerçeve başına işlem süreleri listelenmiştir. Bu sonuçlar Çizelge 5.11'de yer alan, önerilen yöntemin ilk versiyonuna dair sonuçlarla karşılaştırıldığında önemli derecede performans artışı gözlemlenmektedir. Bu performans karşılaştırması Şekil 6.18'de histogram ve autocorrelogram kullanımı için grafiksel olarak gösterilmiştir. Sonuçlar incelendiğinde, yöntemi genişleten sıradüzensel kümeleme ve uyarlanabilir eşikleme bileşenlerinin döngü kapama tespiti maliyetini önemli derecede düşürdüğü görülmektedir.

Çizelge 6.12. Tüm uzaklık metrikleri için, sıradüzensel kümeleme ve uyarlanabilir eşikleme teknikleri ile RGB-D SLAM [44] için çerçeve başına işlem süreleri (ms)

Kayıt	Sıradüzensel Kümeleme				Uyarlanabilir Eşikleme				RGB-D SLAM
	Histogram		Autocorrelogram		Histogram		Autocorrelogram		
	RGB	Gri Ton.	Pik. Uz. = 1	Pik. Uz. = 2	RGB	Gri Ton.	Pik. Uz. = 1	Pik. Uz. = 2	
fr2/360_hemisphere	183.2	175.1	182.4	191.8	177.3	169.7	179.6	188.7	154.1
fr2/coke	178.4	177.7	187.9	196.6	169.1	171.1	175.7	191.9	167.3
fr2/desk	425.1	401.4	417.3	429.8	411.7	396.7	402.4	405.0	348.6
fr2/dishes	143.3	145.7	145.1	159.7	140.8	141.1	146.3	157.7	135.4
fr2/flowerbouquet	234.1	234.9	234.4	246.1	224.8	221.7	231.9	239.6	202.6
fr2/flowerbouquet_br.	172.9	170.9	176.6	189.7	157.3	160.7	174.6	187.8	149.8
fr2/large_no_loop	172.2	164.7	170.1	179.5	166.8	159.4	164.1	174.7	143.9
fr2/large_with_loop	200.0	182.3	186.0	195.1	198.0	177.2	181.7	193.2	156.6
fr2/metallic_sphere	167.0	160.9	168.3	181.5	163.7	155.4	168.0	180.5	155.7
fr2/metallic_sphere2	149.5	147.8	154.0	165.1	142.1	135.9	151.7	162.8	136.4
fr2/pioneer_360	156.9	153.5	159.3	170.6	149.2	146.0	155.0	166.7	139.8
fr2/pioneer_slam	176.7	171.9	182.9	192.7	170.3	168.2	177.3	186.3	155.4
fr2/pioneer_slam2	225.1	218.3	226.3	237.5	216.5	215.9	222.2	230.6	198.9
fr2/pioneer_slam3	170.0	166.9	172.4	183.1	163.6	160.4	168.5	179.8	153.4
Ortalama	196.7	190.9	197.4	208.5	189.4	184.2	192.8	203.2	171.3

Mevcut çerçeve ile başlangıç grubundaki anahtar çerçeveler arasında nitelik eşleştirmesinin *brute-force* yaklaşım (Bkz. Bölüm 5.5.3 ve Bölüm 5.5.4) yerine sıradüzensel kümeleme tekniğiyle gerçekleştirilmesi, yöntemin getirdiği ekstra maliyetin yaklaşık %50 oranında azalmasını sağlamıştır. Ortalamada, *brute-force* yaklaşım ile çerçeve başına işlem süresi yaklaşık %33 oranında artmakta iken, sıradüzensel kümeleme tekniği ile bu oran yaklaşık %16'ya düşmüştür. Bu durum, nitelik eşleştirmesinin sıradüzensel kümeleme ile çok daha hızlı gerçekleştirildiğini göstermektedir. Çünkü *brute-force* yöntemdeki gibi kaynak çerçevedeki her bir anahtar nokta için hedef çerçevedeki tüm noktalar ile mesafe hesaplaması yapılmamaktadır. Performans artışını sağlayan önemli faktörler; kümeleme ağacı tabanlı arama algoritmasının verimliliği ve her bir anahtar çerçeve için sadece bir defa ağaç oluşturularak tüm eşleştirme işlemlerinde aynı ağacın kullanılıyor olmasıdır. Performans artışı *fr2/desk* görüntü kaydı için diğerlerinden çok daha belirgindir çünkü bu kayıt anahtar nokta açısından diğer kayıtlardan çok daha zengindir (Bkz. Şekil 5.15).

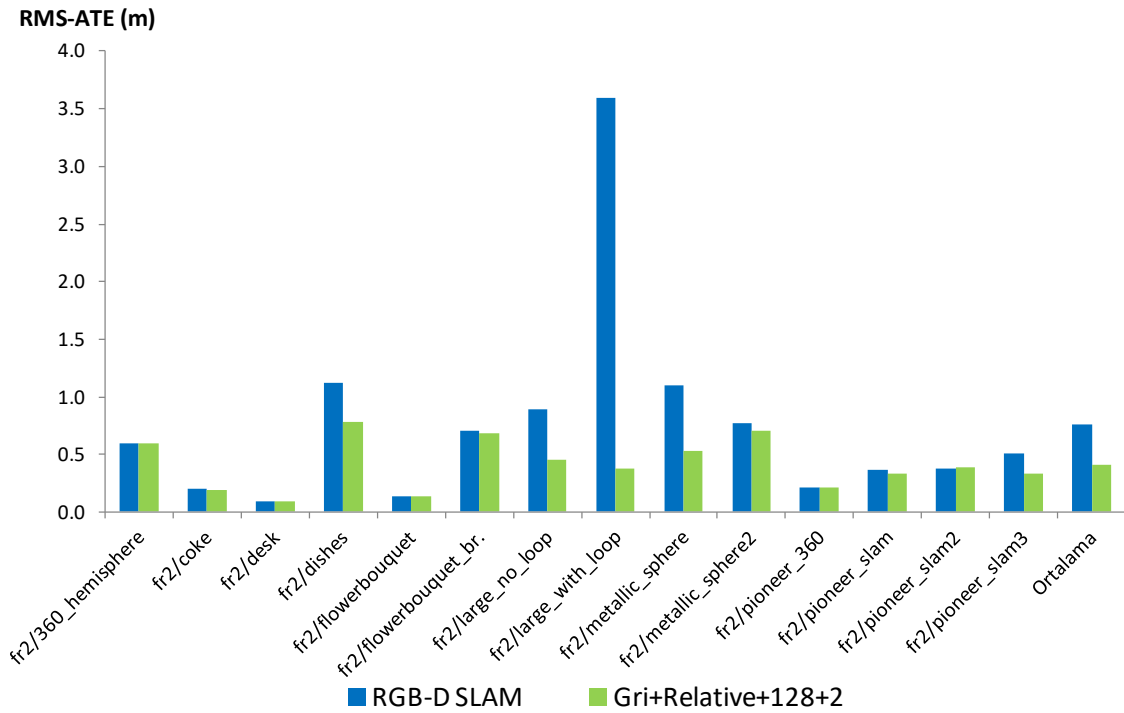


Şekil 6.18. Geliştirilen mekân tanıma tekniklerinin işlemsel maliyet karşılaştırması

Uyarlanabilir eşikleme bileşeninin aktif hale getirilmesi, döngü kapama tespiti yönteminin hızını önemli derecede arttıran bir diğer faktördür. Nitelik eşleştirme maliyetinde önemli bir düşüş sağlayan sıradüzensel kümeleme ile birlikte kullanılan uyarlanabilir eşikleme yaklaşımı, başlangıç grubundan mevcut çerçeveye daha az benzeyen adayları eleyerek nitelik eşleştirme aşamasına daha az sayıda çerçevenin katılmasını sağlamaktadır. Bu sayede bir taraftan modele

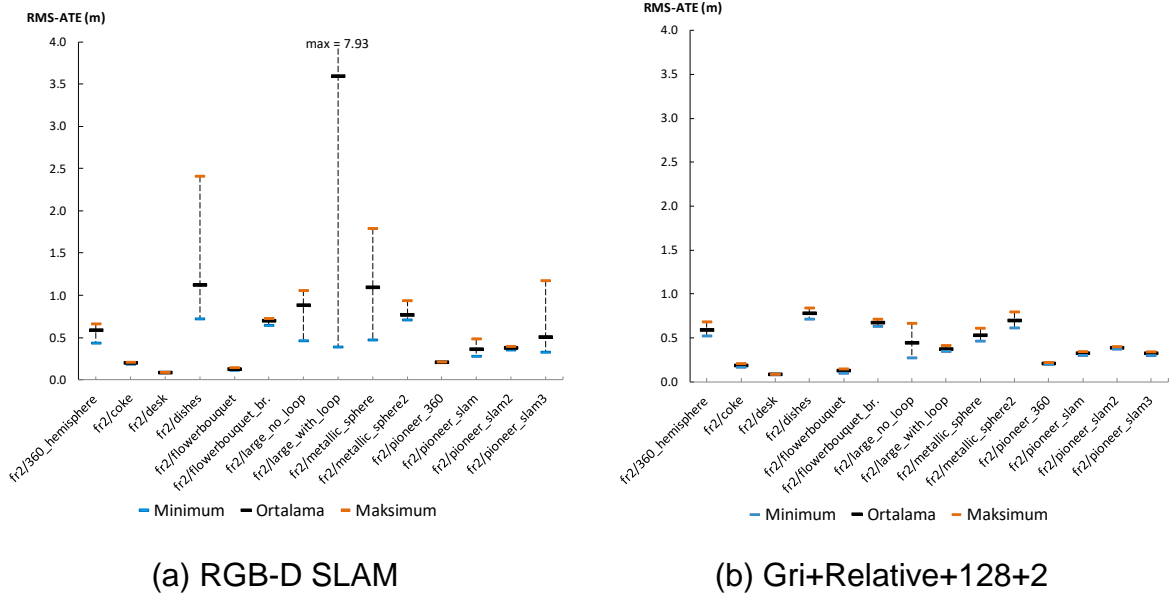
aykırı adaylar elenip daha benzer çerçeveler işleme alınarak yöntemin başarımı korunurken, diğer taraftan ise nitelik eşleştirme işlemine daha az sayıda çerçeve dâhil edilerek önemli bir maliyet düşüşü sağlanmaktadır. *fr2/large\_no\_loop*, *fr2/large\_with\_loop* ve *fr2/pioneer\_slam* serisi gibi geniş robot yolu içeren, daha büyük görüntü kayıtları için performans artışı nispeten daha yüksektir. Bu bağlamda, uyarlanabilir eşikleme tekniği kullanılarak döngü kapama tespiti yönteminin genişletilen şekliyle (tüm bileşenleriyle) uygulanması, genel tanımlayıcı olarak histogram kullanımı için yaklaşık 16 ms, autocorrelogram için ise yaklaşık 27 ms sürmektedir. Dolayısıyla yöntem yaklaşık olarak %12 oranında ekstra maliyet getirmektedir. Sıradüzensel kümeleme tekniğinin yaklaşık %16 oranında işlem maliyetini arttırdığı hesaba katıldığında, uyarlanabilir eşiklemenin %25 civarında maliyeti düşürdüğü görülmektedir.

Netice itibarıyla, önerilen döngü kapama tespiti yönteminin eklenen iki yeni bileşen ile genişletilmesi, genel ve yerel görüntü niteliklerinin birlikte kullanılması durumunda gözlemlenen yaklaşık %33'lük ekstra maliyetin %12 civarına düşmesini sağlamıştır. Bu oran yaklaşık olarak %64 performans artışı anlamına gelmektedir.



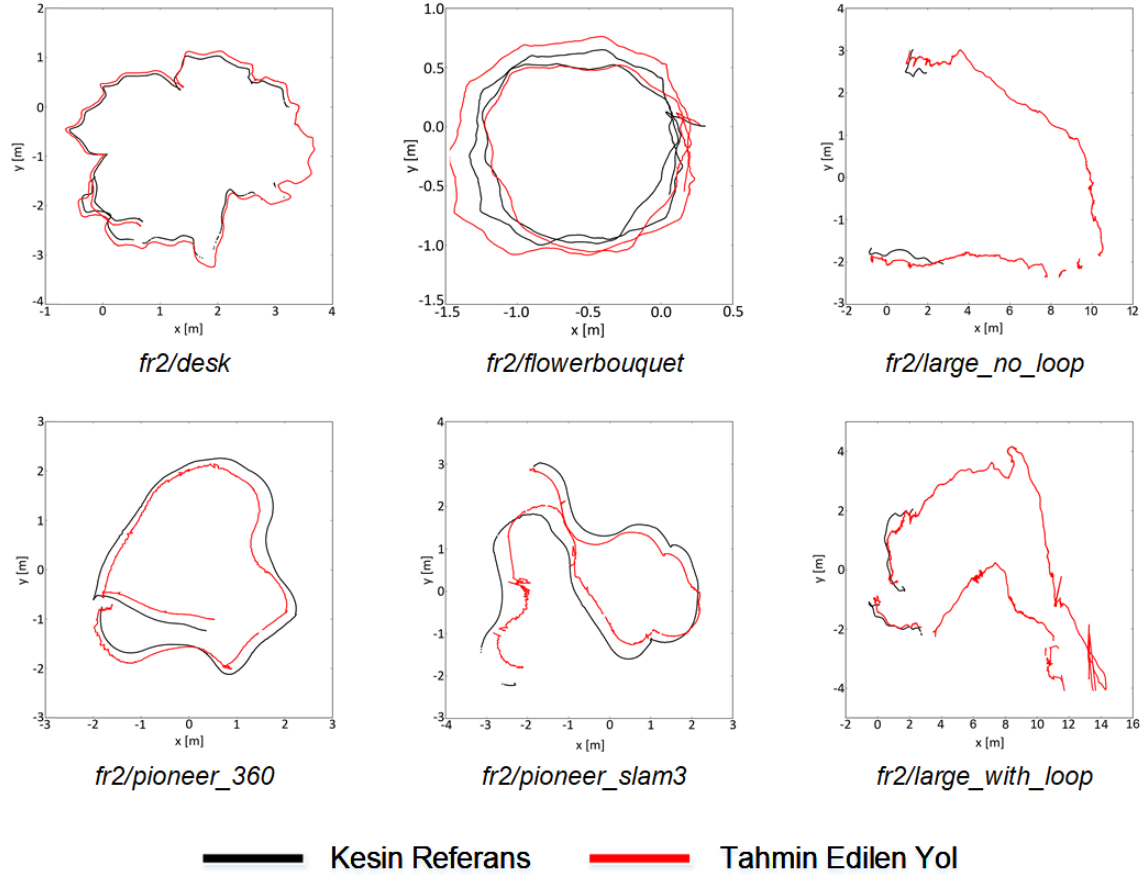
Şekil 6.19. RGB-D SLAM [44] ve önerilen genişletilmiş yöntem (en etkili kombinasyon olan Gri+Relative+128+2) arasında doğruluk karşılaştırması

Şekil 6.19'da, genişletilmiş döngü kapama tespiti yönteminin en etkili kombinasyonu olan Gri+Relative+128+2 ile RGB-D SLAM [44] arasındaki mutlak yol hatası karşılaştırması grafiksel olarak gösterilmektedir. Şekil 6.20'de ise, 5 farklı deneme sonucunda Gri+Relative+128+2 kombinasyonu ve RGB-D SLAM [44] ile elde edilen minimum, ortalama ve maksimum mutlak yol hatası sonuçları ayrıntılı olarak ele alınmıştır. Şekillerden de anlaşılacağı üzere, genişletilmiş yöntemin RGB-D SLAM sistemine [44] entegre edilmesiyle haritalama başarısı önemli derecede yükselmiş ve çok daha stabil sonuçlar elde edilmiştir. RGB-D SLAM [44] için gözlemlenen kayma hatasının ortalama standart sapması 39.5 cm iken, Gri+Relative+128+2 kombinasyonu için bu değer 3.4 cm'dir.



Şekil 6.20. RGB-D SLAM [44] ve en etkili kombinasyon olan Gri+Relative+128+2 için detaylı mutlak yol hatası sonuçları

Şekil 6.21'de yer alan tahmin edilen yol ve kesin referans karşılaştırmaları, döngü kapama tespitinin etkin şekilde gerçekleştirildiğini, dolayısıyla robotun ortamda yapmış olduğu yolun tutarlı şekilde belirlenebildiğini göstermektedir. Özellikle geniş yollar içeren görüntü kayıtlarında elde edilen sonuçlar, genişletilmiş yöntemin hızlı çalışmanın yanında etkili şekilde döngü kapama adaylarını tespit edebildiğine işaret etmektedir.



Şekil 6.21. Genişletilmiş yöntemdeki en etkili kombinasyon olan Gri+Relative+128+2 ile örnek kayıtlar için tahmin edilen yolun ve kesin referansın x-y düzlemine izdüşümü

#### 6.4. Değerlendirme

Bu bölümde, geliştirilen döngü kapama tespiti yöntemi iki yeni bileşen ile genişletilerek döngü kapama adaylarının daha hızlı ve etkili şekilde belirlenmesi sağlanmıştır. Öncelikle, çerçeveler arasında nitelik eşleştirmesi için bir önceki bölümde ifade edildiği gibi *brute-force* yöntemin kullanılması yerine bir yaklaşık en yakın komşu algoritması kullanılmıştır. Sıradüzensel kümeleme adındaki bu yöntem vasıtasıyla her bir anahtar çerçeveye ait nitelik tanımlayıcı vektörleri kümeleme ağacında tutulmakta ve haritalama işlemi boyunca aynı ağaç kullanılmaktadır. Bu sayede önemli bir performans artışı sağlanmıştır. Yönteme yeni eklenen bir diğer bileşen ise, nitelik eşleştirme işleminden önce başlangıç aday grubundaki modele aykırı çerçevelerin, mevcut çerçeveye olan benzerlik sonucuna göre dinamik şekilde elenmesini sağlayan uyarlanabilir eşikleme



yaklaşımıdır. Bu şekilde mevcut çerçeveye daha az benzer adaylar başlangıç grubundan çıkarılarak nitelik eşleştirme işlemine daha düşük sayıda ve daha ilgili anahtar çerçevelerin katılması sağlanmaktadır. Bu teknik ile yöntemin hızı daha da arttırılmış olup, döngü kapama tespiti maliyeti yaklaşık olarak %64 oranında düşürülmüştür. Bunun yanı sıra, yöntem hızlandırılmış haliyle harita doğruluğu açısından herhangi bir dezavantaja sahip olmamış, *brute-force* yöntemine çok yakın haritalama sonuçları elde edilmiştir.

Netice olarak bu bölümdeki çalışma sonucunda, hem genel hem de yerel görüntü niteliklerini birlikte kullanan, sürekli genişleyen arama uzayında döngü kapama adaylarını etkin şekilde belirleyebilen, nitelik tabanlı SLAM sistemlerine kolayca entegre edilebilen, döngü kapamaları başarılı ve hızlı şekilde bulabilen bir döngü kapama tespiti yöntemi geliştirilmiştir. Önerilen yöntemin entegre edilmesiyle genişletilen RGB-D SLAM sistemi, özellikle geniş ölçekli haritalamada başarılı sonuçlar üretmektedir. Döngü kapama tespitini rastgele gerçekleştiren orijinal sisteme göre, çerçeve başına yaklaşık 30 ms ekstra maliyet ile sistem başarımı %46 civarında arttırılmıştır.

## 7. GENİŞ KAPALI ORTAMLAR İÇİN HIZLI RGB-D SLAM YAKLAŞIMI

Bu bölümde, önceki kesimlerde gerçekleştirilmiş olan çalışmalardan edinilen birikimler neticesinde yeni bir RGB-D algılayıcı tabanlı SLAM sistemi geliştirilmiştir. Sistemin ana mekanizmasını anahtar çerçevelerden elde edilen nitelikler oluşturduğundan, geliştirilen yönteme  $k$ -SLAM adı verilmiştir.

Önceki kesimlerde de belirtildiği üzere, literatürde küçük ölçekli ortamlarda etkin şekilde çalışabilen çeşitli RGB-D tabanlı SLAM yöntemleri mevcuttur. Ancak ortam genişledikçe gerek işlem yükünün çok artması, gerek döngü kapama tespitinin çok daha güçleşmesi, gerekse tahmin edilen robot yolundaki kayma miktarının harita tutarlılığını bozacak kadar yükselmesi gibi sebeplerden dolayı haritalama işlemi çok daha zor hale gelmektedir. Bu sebeple, geniş ortamları gerçek zamanlı olarak düşük miktarda hata ile haritalayabilen RGB-D SLAM sistemlerine ihtiyaç vardır.

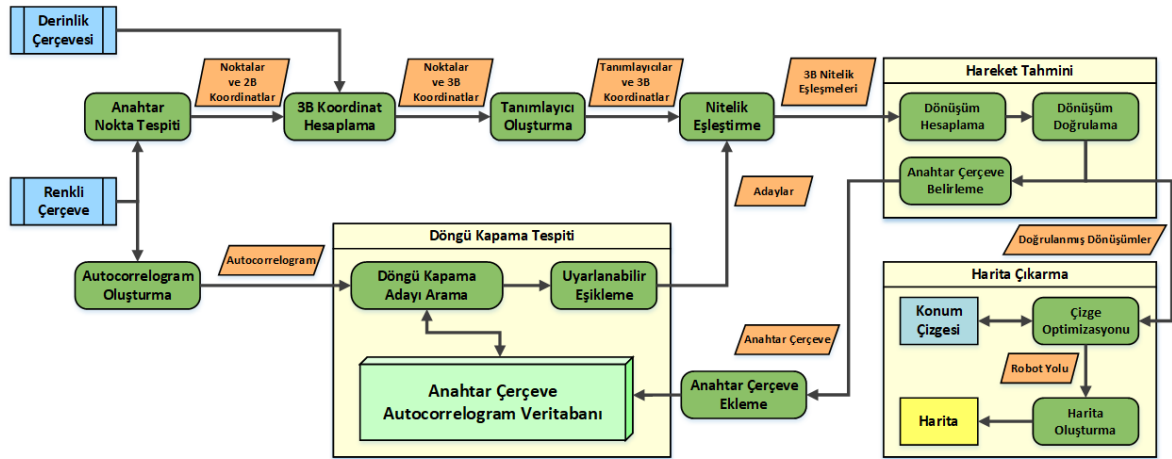
Buradan hareketle, tez kapsamındaki önceki çalışmalardan elde edilen sonuçlardan faydalanılarak hem küçük hem de geniş ortamlarda gerçek zamanlı çalışabilen bir SLAM sistemi tasarlanmıştır. Yaygın olarak kullanılan veri kümeleri üzerinde sistem test edilmiş ve literatürdeki diğer gelişmiş SLAM yöntemleriyle karşılaştırma yapılmıştır. Deneysel sonuçlar,  $k$ -SLAM sisteminin zorlu şartlar içeren geniş ortamların haritalanmasında etkili sonuçlar üretebildiğini göstermektedir. Ayrıca küçük ortamlarda da robot hareketi düşük hata ile tahmin edebilmektedir.  $k$ -SLAM sistemi, kullandığı etkin döngü kapama tespiti yöntemi sayesinde geniş ortamlarda CPU üzerinde gerçek zamanlı olarak çalışabilmektedir.

### 7.1.Yöntem

#### 7.1.1. $k$ -SLAM Sistemine Genel Bakış

Geliştirilen 3B SLAM sistemi, günümüzde yaygın olarak kullanılan çizge tabanlı yaklaşımı benimsemektedir. Algılayıcıdan gelen her yeni çerçeveye için çizgeye yeni bir düğüm ekleyen sistem, görüntü çerçeveleri arasındaki robot hareketinin tahmin edilmesi için nitelik tabanlı çakıştırma yaklaşımını kullanmaktadır. Döngü kapama tespiti, genel görüntü niteliklerinin bir veri yapısı içerisinde indekslenmesine dayalı olarak gerçekleştirilmektedir. Genel görüntü niteliği olarak autocorrelogram yapısı kullanılmış olup, görüntüdeki renk / gri tonu seviyesi dağılımına dair uzamsal

ilişkilerden faydalanılmıştır. Görüntü autocorrelogramları, bir öncelik araması k-ortalamlar ağacı (*priority search k-means tree*) [89] içerisinde indekslenmektedir. Anahtar Çerçeve Autocorrelogram Veritabanı (*Keyframe Autocorrelogram Database*) adı verilen bu yapı, döngü kapama adaylarının hızlı ve başarılı şekilde belirlenmesini sağlamaktadır. Bildiğimiz kadarıyla *k*-SLAM sistemi, SLAM literatüründe döngü kapama tespiti için görüntü autocorrelogramları ile bu şekilde bir indeksleme yapısını kullanan ilk sistemdir. Diğer yandan, modele aykırı döngü kapama adaylarının elenmesi için daha önceki kesimde geliştirilmiş olan uyarlanabilir eşikleme tekniği kullanılmıştır. Sistemdeki döngü kapama tespiti bileşeni, ortam genişliğinden bağımsız olarak herhangi bir eğitim işlemi gerektirmemektedir. Tüm görüntü çerçeveleri işlendikten sonra konum çizgesi optimize edilmekte ve ortam haritası oluşturulmaktadır. Sistemin ana işleyişi Şekil 7.1’de yer almaktadır.



Şekil 7.1. *k*-SLAM sisteminin genel mimarisi

### 7.1.2. Nitelik Tabanlı Hareket Tahmini

Sistem çerçeveler arasındaki robot hareketini, anahtar noktalar üzerinden aralarındaki dönüşümü hesaplayarak tahmin etmektedir. Çerçevelerden nitelik çıkarımı için, önceki çalışmalardan hareketle hız ve doğruluk açısından en başarılı kombinasyon tercih edilmiştir. Dolayısıyla, çerçevelerdeki anahtar noktaların tespit edilmesi için CenSurE nitelik bulucusu, noktalara ait tanımlayıcıların oluşturulması için ise FREAK nitelik tanımlayıcısı kullanılmıştır.

Öncelikle, renkli çerçevelerdeki anahtar noktalar tespit edilmekte ve noktalara ait tanımlayıcı vektörler elde edilmektedir. Tespit edilen noktalara dair 3B koordinatlar, ilgili derinlik çerçevelerinden yararlanılarak hesaplanmaktadır. Bu aşamada, geçerli derinlik bilgisine sahip olmayan (algılayıcının menzil kısıtı ya da gürültü gibi sebeplerden kaynaklı) anahtar noktalar elenmektedir. Sonrasında ise Hamming mesafesi kullanılarak çerçeveler arasında nitelik eşleştirmesi gerçekleştirilmekte ve anahtar nokta eşleşmeleri 3B koordinatlarıyla birlikte elde edilmektedir. Yanlış eşleşmelerin elenmesi için, önceki çalışmalarda da kullanılmış olan uzaklık oranı yaklaşımı [24] uygulanmaktadır (Bkz. Bölüm 5.3.2). Dolayısıyla herhangi bir anahtar nokta eşleşmesinin geçerli sayılabilmesi için, kaynak çerçevedeki noktaya hedef çerçevede karşılık gelen en yakın iki nokta için hesaplanan uzaklıklar arasındaki oranın bir eşik değerden (*oran\_eşiği*) yüksek olması gereklidir (Bkz. Eşitlik 5.11). *oran\_eşiği* parametresi için, ayrıntılı deneysel analizlerden hareketle 0.8 değeri uygulanmıştır.

Nitelik eşleştirmesi işleminden sonra, elde edilen 3B eşleşmeler üzerinde RANSAC algoritması uygulanarak çerçeveler arasındaki hareket tahmin edilmektedir. Her bir RANSAC yinelemesinde, Arun ve diğerlerinin [90] yöntemine benzer şekilde tekil değer ayrışımı (*singular value decomposition*) kullanılarak dönüşüm hesaplaması yapılmaktadır. Eşleşen anahtar noktalar arasındaki Mahalanobis uzaklığına göre eşleşmelerin modele uygun olup olmadığına karar verilmektedir. RANSAC algoritması uygulandıktan sonra elde edilen dönüşüm üzerinde, yeterli sayıda modele uygun değer içerip içermediğine göre doğrulama gerçekleştirilmekte ve doğrulanamayan dönüşümler elenmektedir.

### **7.1.3. Döngü Kapama Tespiti**

Döngü kapamaların başarılı ve hızlı şekilde tespit edilebilmesi için, etkili bir genel tanımlayıcı olan autocorrelogram yapısı sıradüzensel kümeleme yoluyla kullanılmıştır. Öncelik araması yoluyla döngü kapama adaylarını tespit eden bu bileşenin yanında, uyarlanabilir eşikleme tekniği uygulanarak sistem performansı arttırılmaktadır.

#### **7.1.3.1. Autocorrelogram Kullanımı**

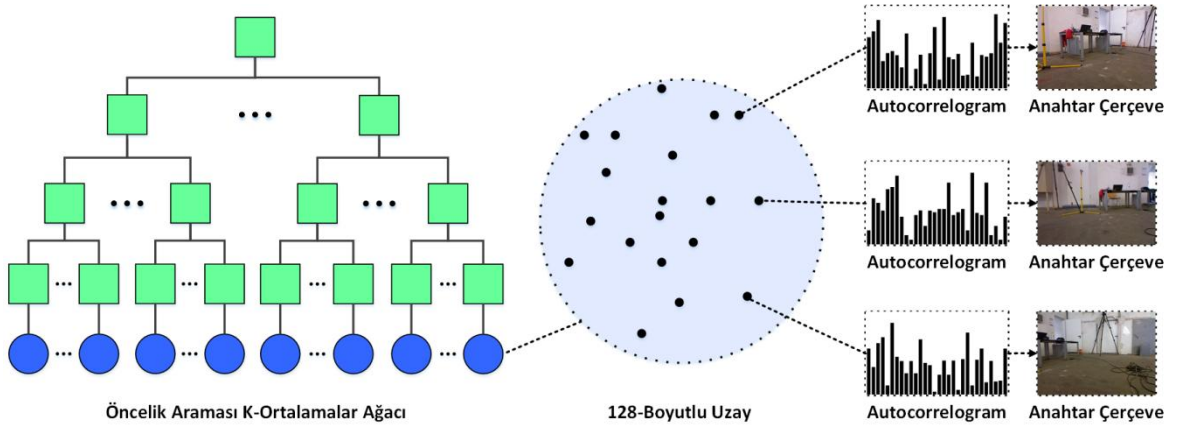
Önceki çalışmalardan hareketle, daha başarılı performans sergileyen autocorrelogram yapısı genel tanımlayıcı olarak sisteme entegre edilmiştir

(Bkz. Bölüm 5.3.1.2). Hem içerdiği zengin veri hem de kolayca hesaplanabilmesi ve kıyaslanabilmesi yönleriyle görüntü autocorrelogramları sistem tarafından döngü kapama tespitinde etkin şekilde kullanılmaktadır.

Algılayıcıdan gelen her bir renkli çerçeve sistem tarafından gri tonlamalı şekle dönüştürülmekte ve 128 gri tonu seviyesine nicelendirilmektedir. Sonrasında ise çerçeveye ait autocorrelogram 1 piksel uzaklığına göre hesaplanmaktadır. Döngü kapama adayları, mevcut çerçeve ile anahtar çerçeveler arasındaki autocorrelogram benzerliklerine göre bir autocorrelogram veri tabanında (sıradaki kesimde açıklanmaktadır) aranmaktadır. Autocorrelogram benzerliğinin ölçülmesi için Manhattan uzaklığı kullanılmaktadır (Bkz. Eşitlik 5.9). Kullanılan autocorrelogram parametrelerinin ve uzaklık metriğinin seçiminde doğruluğun yanında hız faktörü de göz önünde bulundurulmuştur.

### 7.1.3.2. Anahtar Çerçeve Autocorrelogram Veritabanı

Sistem döngü kapama tespiti için bir Anahtar Çerçeve Autocorrelogram Veritabanı kullanılmaktadır. Bu veritabanı, anahtar çerçevelere ait autocorrelogramları içeren bir öncelik araması k-ortalamlar ağacından [89] oluşmaktadır. Şekil 7.2’de veritabanının detaylı yapısı görülmektedir.



Şekil 7.2. Anahtar Çerçeve Autocorrelogram Veritabanı'nın yapısı

Döngü kapama tespiti için anahtar çerçeve mekanizması kullanılmıştır, çünkü algılayıcıdan gelen her bir yeni çerçeve için önceki çerçevelerin tamamının işlenmesi maliyetin aşırı derecede artmasına sebep olmaktadır. Dolayısıyla döngü kapama adaylarının robot yolunu içeren çerçeveler kümesinin tamamında aranması yerine, anahtar çerçeveler kullanılarak arama maliyeti azaltılmaktadır.

Öncelik araması  $k$ -ortalamalar ağacı, herhangi bir noktanın yaklaşık en yakın komşularını yüksek hassasiyetle elde edebilen bir yapıdır. Bu ağaç yapısında noktalar tüm boyutlardaki uzaklığa göre kümelendirilmektedirler. Sistemde, her bir anahtar çerçeve autocorrelogramı ayrı bir nokta olarak bu ağaçta tutulmaktadır. Autocorrelogramlar 128 gri tonu seviyesi ve 1 piksel uzaklığına göre oluşturulduğundan, her bir autocorrelogram 128 adet kayan noktalı sayıdan oluşan bir vektör anlamına gelmektedir. Dolayısıyla her bir anahtar çerçeve autocorrelogramı 128 boyutlu uzayda bir nokta olarak ele alınmaktadır (Bkz. Şekil 7.2).

Öncelik araması  $k$ -ortalamalar ağacı, 128 boyutlu arama uzayının sıradüzensel şekilde parçalara ayrılması yoluyla oluşturulmaktadır. Ağaçtaki ara düğümler küme merkezlerini, yaprak düğümler ise noktaları içerirler. Ağacın oluşturulması için kullanılan temel parametreler dallanma faktörü  $k$  ve maksimum yineleme sayısı  $I_{max}$ 'tir. Autocorrelogramları ifade eden noktalar her bir seviyede  $k$ -ortalamalar kümelemesi (*k-means clustering*) ile özyineli olarak  $k$  bölgeye ayrılmaktadırlar. Her bir seviyede,  $k$  adet küme merkezi rastgele olarak belirlenmekte ve her bir nokta Manhattan uzaklığına göre en yakın kümeyle ilişkilendirilmektedir. Sonrasında ise küme merkezleri, yakınsamaya ya da maksimum yineleme sayısına ulaşıncaya kadar küme ortalamalarına göre tekrarlı olarak ayarlanmaktadır. Kümeler belirlendikten sonra merkez noktalar için ara düğümler,  $k$ 'dan az sayıda nokta içeren kümeler için ise yaprak düğümler oluşturulmaktadır. Tüm kümeler  $k$ 'dan düşük sayıda nokta içerinceye kadar bu işlem özyineli olarak uygulanmaktadır.

---

**Input** : öncelik araması k-ortalamlar ağacı  $T$ , autocorrelogram  $A$ , dallanma faktörü  $k$ , maksimum yineleme sayısı  $I_{max}$ , yeniden oluşturma eşiği  $r$

**Output**: yeni autocorrelogramın eklendiği öncelik araması k-ortalamlar ağacı  $T$

$S_p \leftarrow$  son yeniden oluşturma işleminde  $T$ 'nin büyüklüğü  
 $S_c \leftarrow T$ 'nin mevcut büyüklüğü

**if**  $(S_p \times r) < (S_c + 1)$  **then**  
     $N \leftarrow T$ 'deki tüm autocorrelogramlar  
     $N \leftarrow N + A$  { $A$ 'yı autocorrelogramlar kümesine ekle}  
    call AutocorrelogramlariKumele( $N, k, I_{max}$ )  
**else**  
     $Z \leftarrow T$ 'nin kök düğümü  
    call AutocorrelogramiAgacaEkle( $Z, A, k, I_{max}$ )  
**end**  
**return** öncelik araması k-ortalamlar ağacı

**Procedure** AutocorrelogramiAgacaEkle( $Z, A, k, I_{max}$ )  
    **if**  $Z$  yaprak düğüm **then**  
        autocorrelogram  $A$ 'yı düğüm  $Z$ 'ye ekle  
        **if**  $|Z| \geq k$  **then**  
             $D \leftarrow$  düğüm  $Z$ 'deki tüm autocorrelogramlar  
            call AutocorrelogramlariKumele( $D, k, I_{max}$ )  
        **end**  
    **else**  
         $Z_c \leftarrow Z$ 'nin  $A$ 'ya en yakın alt düğümünü bul  
        call AutocorrelogramiAgacaEkle( $Z_c, A, k, I_{max}$ )  
    **end**  
**end**

**Procedure** AutocorrelogramlariKumele( $N, k, I_{max}$ )  
    **if**  $|N| < k$  **then**  
         $N$ 'deki autocorrelogramlarla yaprak düğüm oluştur  
    **else**  
         $F \leftarrow N$ 'den küme merkezi olarak  $k$  adet autocorrelogramı rastgele seç  
        yakınsama  $\leftarrow$  false  
        yineleme  $\leftarrow 0$   
        **while** yakınsama yok and yineleme  $< I_{max}$  **do**  
             $C \leftarrow N$ 'deki autocorrelogramları  $F$ 'deki en yakın merkezler etrafında kümele  
             $F_{mean} \leftarrow C$ 'deki kümelerin ortalamaları  
            **if**  $F = F_{mean}$  **then**  
                yakınsama  $\leftarrow$  true  
            **end**  
             $F \leftarrow F_{mean}$   
            yineleme  $\leftarrow$  yineleme + 1  
        **end**  
        **for** her bir küme  $C_i \in C$  **do**  
             $F_i$  merkezli ara düğümü oluştur  
            algoritmayı  $C_i$ 'deki autocorrelogramlara özyineli olarak uygula  
        **end**  
    **end**  
**end**

---

Şekil 7.3. Öncelik araması k-ortalamlar ağacını artırımsal olarak oluşturma

Sistemde öncelik araması k-ortalamlar ağacı artırimsal olarak oluşturulmaktadır (Bkz. Şekil 7.3). Algılayıcıdan gelen ilk çerçeve, ilk anahtar çerçeve olarak nitelendirilmekte ve bu çerçeveye ait autocorrelogram ilk nokta olarak kabul edilerek ağaç oluşturulmaktadır. Haritalama esnasında yeni anahtar çerçeveler belirlendikçe, ilgili yeni autocorrelogramlar ağaca artırimsal olarak eklenmektedir. Ağaçtaki dengenin bozulmasının engellenmesi için, ağaçtaki nokta sayısının iki katına çıktığı her seferde, mevcut tüm anahtar çerçeve autocorrelogramları kullanılarak önceki paragrafta açıklanan yöntemle ağaç yeniden oluşturulmaktadır. Dolayısıyla ağaç için yeniden oluşturma eşiği ( $r$ ) 2 olarak belirlenmiştir. Hassasiyet ve verimlilik arasındaki dengenin sağlanması için, dallanma faktörü  $k$  değeri 32 olarak, maksimum yineleme sayısı  $I_{max}$  ise 11 şeklinde uygulanmıştır.

---

**Input** : öncelik araması k-ortalamlar ağacı  $T$ , autocorrelogram  $A$ , aday sayısı  $K$ , kontrol edilen maksimum autocorrelogram sayısı  $C_{max}$

**Output**: döngü kapama adayları (autocorrelogram  $A$ 'nın yaklaşık en yakın  $K$  komşusu)

$C \leftarrow 0$  {kontrol edilen autocorrelogram sayısını tutar}

$PQ \leftarrow$  ele alınmayan düğümler için boş öncelik kuyruğu

$R \leftarrow$  bulunan komşular için boş öncelik kuyruğu

call KOrtalamlarAgaciniDolas( $T, PQ, R, C, A$ )

**while**  $|PQ| > 0$  and  $C < C_{max}$  **do**

$D \leftarrow PQ$  kuyruğundaki en üst düğüm

    call KOrtalamlarAgaciniDolas( $D, PQ, R, C, A$ )

**end**

**return**  $R$ 'de en üstte yer alan  $K$  autocorrelogram

**Procedure** KOrtalamlarAgaciniDolas( $D, PQ, R, C, A$ )

**if**  $D$  yaprak düğüm **then**

$D$ 'deki tüm autocorrelogramları  $R$ 'ye ekle

$C \leftarrow C + |D|$

**else**

$N \leftarrow D$ 'nin alt düğümleri

$N_q \leftarrow N$ 'de yer alan, autocorrelogram  $A$ 'ya en yakın düğüm

$N_p \leftarrow N - N_q$

$N_p$ 'deki tüm düğümleri  $PQ$ 'ya ekle

        call KOrtalamlarAgaciniDolas( $N_q, PQ, R, C, A$ )

**end**

**end**

---

Şekil 7.4. Anahtar Çerçeve Autocorrelogram Veritabanı'nda döngü kapama adayı arama

Algılayıcıdan gelen her bir çerçeve için döngü kapama adayları, Anahtar Çerçeve Autocorrelogram Veritabanı'nda arama yapılarak belirlenmektedir. Bunun için öncelik araması k-ortalamlar ağacı özyineli olarak dolaşmaktadır (Bkz. Şekil



7.4). Daha önce değinildiği üzere, ağaç dolaşımı sırasında autocorrelogram karşılaştırmaları Manhattan uzaklık metriği kullanılarak gerçekleştirilmektedir. Arama kök düğüm ile başlamakta ve dolaşımın her bir özyinelemesinde mevcut çerçeveye ait autocorrelograma en yakın küme merkezini içeren düğüm seçilmektedir. Yol üzerinde ele alınmayan düğümler, uzaklıklarına göre bir öncelik kuyruğuna eklenmektedir. Yaprak düğüme ulaşıldıktan sonra, bu kümede yer alan autocorrelogramlar mevcut düğüme ait autocorrelogramla karşılaştırılmakta ve en yakın komşular seçilmektedir. Kümede yeterli sayıda autocorrelogram yoksa, öncelik kuyruğundan en yakın düğüm seçilmekte ve dolaşım bu düğümden devam etmektedir.

Bu şekilde, döngü kapama tespitinin ilk adımı olarak, anahtar çerçeve autocorrelogramlarından oluşan bir kümeleme ağacında arama yapılması ve mevcut çerçeveye autocorrelogram benzerliği yönünden en yakın komşuların bulunması yoluyla belirli sayıda döngü kapama adayı elde edilmektedir.

#### **7.1.3.3. Uyarlanabilir Eşikleme**

Her bir döngü kapama aramasında sabit sayıda döngü kapama adayının seçilmesi bazı ilgisiz anahtar çerçevelerin (modele aykırı adaylar) de işleme katılmasına sebep olabilmektedir. Bu durum hem kayma hatasını hem de işlemsel yükü arttırabilen bir faktördür. Dolayısıyla, mevcut çerçeveye daha düşük benzerliğe sahip anahtar çerçevelerin elenmesi için, önceki çalışmada geliştirilmiş olan uyarlanabilir eşikleme mekanizması (Bkz. Bölüm 6.1.2) kullanılmaktadır. Eleme işlemi, Anahtar Çerçeve Autocorrelogram Veritabanı araması ile belirlenen aday grubu üzerinde, autocorrelogramlar arasındaki Manhattan uzaklığına göre dinamik bir eşik değerin uygulanmasıyla gerçekleştirilmektedir.

Daha önce de değinildiği üzere, her döngü kapama aramasında sabit bir eşik değerin kullanılmasıyla etkili performans sağlanamamaktadır çünkü benzerlik skorları arama çerçeveleri bazında değişmektedir. Algılayıcıdan gelen her yeni çerçeve farklı bir görsel karakteristiğe sahiptir. Dolayısıyla aynı eşik değerin uygulanması doğru döngü kapama adaylarının elenmesine sebep olabilir. Buradan hareketle benzerlik eşiği, arama çerçevesi ile aday grubu içerisindeki en benzer anahtar çerçeve arasındaki autocorrelogram benzerliğine göre dinamik olarak hesaplanmaktadır.

Anahtar Çerçeve Autocorrelogram Veritabanı araması sonucunda elde edilen her bir aday çerçeve, mevcut çerçeve ile arasındaki autocorrelogram benzerliğine göre değerlendirilerek, o anki döngü kapama araması için belirlenmiş olan dinamik eşik değerden düşük benzerliğe sahipse elenmektedir.



Şekil 7.5. Uyarlanabilir eşikleme tekniği ile modele aykırı adayların elenmesine dair örnekler

Uyarlanabilir eşikleme mekanizması, her bir döngü kapama aramasında önemli sayıda ilgisiz adayı eleyerek hem haritalama doğruluğunun artırılmasını hem de işlem maliyetinin düşürülmesini sağlamaktadır. Şekil 7.5'te örnek döngü kapama aramaları için uyarlanabilir eşikleme ile elde edilen sonuçlar görülmektedir. Burada *eşikleme\_faktörü* parametresi 1.5 olarak uygulanmış olup, döngü kapama tespitinin ilk aşaması olan Anahtar Çerçeve Autocorrelogram Veritabanı araması ile 15 çerçevelik aday grubu seçilmektedir. Şekilde de görüldüğü üzere, elenen aday sayıları farklı döngü kapama aramaları için dinamik olarak değişmektedir. *eşikleme\_faktörü* parametresinin değeri değiştirilerek modele uygun - modele aykırı aday dengesi ayarlanabilir. Düşük eşik değerlerin kullanılması daha fazla adayın elenmesini sağlayarak işlemsel maliyeti daha da azaltabilir, ancak modele uygun adayların elenme olasılığı artmaktadır.

#### 7.1.3.4. Robot Yolu Tahmini ve Harita Oluşturulması

Sistem robotun yaptığı yolu, algılayıcıdan gelen çerçevelerle konum çizgesini yeni düğüm ve kenarlar ile genişleterek tahmin etmektedir. Her yeni çerçeve için çizgeye yeni bir düğüm eklenmekte ve hareket tahmini sonuçlarına göre bu düğümle diğer düğümler arasında yeni kenarlar oluşturulmaktadır. Yeni düğümle

aralarında hareket tahmini yapılacak olan düğümler, belirli sayıda en son gelen çerçeve ve döngü kapama adaylarıdır. Her yeni gelen çerçeve için ilk olarak nitelik çıkarımı gerçekleştirildiğinden dolayı, çerçeveler arasında nitelik eşleştirme aşamasında zaten mevcut olan tanımlayıcılar kullanılmaktadır. Sonrasında ise yeni gelen çerçeve ile seçilen çerçevelerin her biri arasında dönüşüm hesaplaması gerçekleştirilmektedir. Doğrulan her dönüşüm için, çizgede ilgili düğümler arasında odometri kısıtlarını (direkt önceki çerçeveler için) ya da döngü kapama kısıtlarını (döngü kapama adayları için) içeren yeni kenar oluşturulmaktadır. Ayrıca, yeni gelen çerçeve ile hiçbir anahtar çerçeve arasında kenar oluşturulamazsa, mevcut çerçeveden bir önceki çerçeve yeni anahtar çerçeve olarak belirlenmekte ve Anahtar Çerçeve Autocorrelogram Veritabanı'na eklenmektedir. Bu şekilde konum çizgesi yeni düğümlerin ve doğrulan dönüşümler için yeni kenarların eklenmesiyle genişlemektedir. Tüm çerçeveler işlendikten sonra çizge g2o yöntemiyle [37] optimize edilmektedir. Optimizasyon sonrasında elde edilen robot yolu kullanılarak ortamın haritası oluşturulmaktadır. Robot yolundaki her bir konumda elde edilmiş olan algılayıcı gözlemleri ortak bir koordinat sistemine yansıtılarak ortamın bir nokta bulutu gösterimi oluşturulmaktadır.

## 7.2. Deneysel Kurulum

*k*-SLAM sistemi, bu alanda popüler olarak kullanılan ICL-NUIM ve TUM RGB-D Benchmark veri kümeleri üzerinde test edilmiş ve literatürdeki en başarılı RGB-D tabanlı haritalama sistemleri ile karşılaştırılmıştır. Deneylerde; küçük, orta ölçekli ya da geniş ortamlarda kaydedilme, kısa ya da uzun robot yolları içermeye, zorlu ya da nispeten daha basit sahnelere sahip olma gibi çok çeşitli koşullar içeren büyük bir görüntü kaydı kümesi kullanılmıştır.

Gerçekleştirilen deneylerde, döngü kapama tespiti için ilk adımda Anahtar Çerçeve Autocorrelogram Veritabanı'ndan en fazla 15 aday seçilmektedir. Uyarlanabilir eşikleme aşamasında ise *eşikleme\_faktörü* parametresi için 2.0 değeri uygulanarak nihai döngü kapama adayları belirlenmektedir. Mevcut çerçeve ile aralarında hareket tahmini yapılacak olan direkt önceki çerçevelerin sayısı, orta ölçekli görüntü kayıtları için 5 gibi düşük sayılarda seçilmekte, geniş kayıtlar için ise 20'ye kadar çıkarılmaktadır. Deneyler, önceki çalışmalarda da olduğu gibi

Ubuntu 12.04 işletim sistemine sahip, Intel Core i7-2600 CPU 3.40GHz işlemci ve 8GB RAM bulunduran bir bilgisayar üzerinde gerçekleştirilmiştir.

### 7.3. Sonuçlar

Bu kesimde,  $k$ -SLAM sisteminin test edildiği görüntü kayıtları ile elde edilen sonuçlar doğruluk ve hız açısından incelenmiş ve diğer sistemlerle kıyaslama yapılmıştır.

#### 7.3.1. ICL-NUIM Veri Kümesi Deneyleri

Geliştirilen sistem ilk olarak, daha küçük ortamlarda kaydedilmiş ve nispeten daha kolay koşullara sahip olan ICL-NUIM veri kümesindeki görüntü kayıtları üzerinde denenmiştir. Bu kayıtlar kısa yollar içermekte ve döngü kapama senaryolarına pek sahip değildir. Dolayısıyla sistemin performansını temel olarak odometri tahmini belirlemektedir. Çizelge 7.1’de, ICL-NUIM veri kümesindeki kayıtlara dair ayrıntılı bilgiler yer almaktadır.

Çizelge 7.1. Deneylerde kullanılan ICL-NUIM görüntü kayıtları hakkında ayrıntılı bilgiler

Sahne	Kayıt	Süre (s)	Yol Uzunluğu (m)	Ort. Öteleme Hızı (m/s)
Living Room	kt0	50.3	6.54	0.126
	kt1	32.2	2.05	0.063
	kt2	29.4	8.43	0.282
	kt3	41.4	11.32	0.263
Office Room	kt0	50.3	6.54	0.126
	kt1	32.2	6.73	0.206
	kt2	29.4	9.02	0.302
	kt3	41.4	7.83	0.182

Çizelge 7.2 ve Çizelge 7.3’te, ICL-NUIM veri kümesindeki sırasıyla gürültü içermeyen ve gürültülü görüntü kayıtları ile elde edilen doğruluk sonuçları yer almaktadır. Geliştirilen  $k$ -SLAM sisteminin karşılaştırıldığı literatürdeki gelişmiş yöntemler; Steinbrücker ve diğerlerinin [49] enerji tabanlı yöntemi (RGB-D olarak gösterilmiştir), Newcombe ve diğerlerinin [54] metodunda olduğu gibi ICP’nin kesilmiş işaretli uzaklık fonksiyonu ile birlikte kullanılması, Whelan ve diğerlerinin [56] Kintinuous sistemi (ICP+RGB-D kombinasyonu ile), Kerl ve diğerlerinin [50] yoğun görsel odometri (*dense visual odometry*, DVO) yaklaşımı, Stückler ve

Behnke [59] tarafından geliştirilen MRSSMap ve Whelan ve diğerlerinin [60] ElasticFusion sistemi şeklindedir.

Çizelge 7.2. ICL-NUIM veri kümesinde yer alan gürültüsüz kayıtlar üzerinde mutlak yol hatası (RMS-ATE) karşılaştırması (m)

Sistem	Living Room				Office Room			
	kt0	kt1	kt2	kt3	kt0	kt1	kt2	kt3
RGB-D	0.4558	0.6288	0.1609	1.0294	0.2701	0.6173	0.2664	0.4750
ICP	0.1188	<b>0.0023</b>	<b>0.0015</b>	0.0200	<b>0.0029</b>	<b>0.0385</b>	<b>0.0016</b>	<b>0.0021</b>
Kintinuous	0.4365	0.0096	0.2151	0.6975	0.2248	0.4558	0.6367	0.0535
DVO	0.1138	0.1055	0.1073	0.1879	0.3977	0.4461	0.3271	0.2066
<i>k</i> -SLAM	<b>0.0370</b>	0.0070	0.0136	<b>0.0104</b>	0.0094	0.1286	0.0117	0.0102

Çizelge 7.3. ICL-NUIM veri kümesinde yer alan simüle edilmiş gürültü içeren kayıtlar üzerinde mutlak yol hatası (RMS-ATE) karşılaştırması (m)

Sistem	Living Room				Office Room			
	kt0	kt1	kt2	kt3	kt0	kt1	kt2	kt3
RGB-D	0.3603	0.5951	0.2931	0.8524	0.1710	0.5366	0.2289	0.2298
ICP	0.0724	<b>0.0054</b>	<b>0.0104</b>	0.3554	<b>0.0216</b>	0.9691	<b>0.0109</b>	0.9323
Kintinuous	0.3936	0.0214	0.1289	0.8640	0.2495	0.4395	0.4750	0.0838
DVO	0.2911	0.1246	0.4733	0.5436	0.3350	0.3778	0.3593	0.2338
MRSSMap	0.2040	0.2280	0.3310	1.1270	-	-	-	-
ElasticFusion	<b>0.0090</b>	0.0090	0.0140	0.1060	-	-	-	-
<i>k</i> -SLAM	0.0152	0.0076	0.0155	<b>0.0178</b>	0.0240	<b>0.1206</b>	0.0160	<b>0.0152</b>

Elde edilen sonuçlar, *k*-SLAM'in en gelişmiş haritalama sistemlerinin çoğunluğundan daha başarılı performans sergilediğini göstermektedir. Görüntü kayıtlarının neredeyse tamamında *k*-SLAM ilk iki sırada yer almakta, en iyi sonucu üretmekte ya da ICP yönteminin ardından gelmektedir. Geliştirilen sistemin ürettiği kayma hatası, gürültülü kayıtlar da dâhil olmak üzere çoğunlukla 1 cm civarındadır. Çerçeveler üzerinde elde edilen anahtar nokta sayısının artmasıyla mutlak yol hatasının azaldığı gözlenmektedir (ortalama anahtar nokta sayıları için Bkz. Şekil 7.7), çünkü sahnedeki dokunun daha fazla olması daha başarılı hareket tahmini yapılabilmesine imkân vermektedir. Veri kümesindeki görüntü kayıtları, *k*-SLAM gibi nitelik tabanlı bir sistem için nispeten düşük sayıda anahtar nokta içermelerine rağmen, kayıtlardaki düşük öteleme hızı (Bkz. Çizelge 7.1) dönüşüm hesaplamasındaki hatanın azalmasını sağlayan önemli bir faktördür. Diğer görüntü

kayıtlarına nazaran, *Office Room* grubundaki *kt1* kaydı için daha yüksek kayma hataları elde edilmiştir. Bu durumun sebebi, görüntü kaydının duvara odaklanmaktan kaynaklanan düşük doku ve geometrik yapılı bir bölüm içermesidir. Görüntü kayıtlarına simüle edilmiş algılayıcı gürültüsünün eklenmesi, ICP yöntemi için özellikle *Office Room* grubundaki kayıtlarda hatayı büyük miktarda arttırmaktadır. Bu durum, çerçeve çakıştırma işlemi için sadece derinlik verisi kullanılmasının gürültüden önemli derecede etkilendiğini göstermektedir. Gürültü içeren ve gürültüsüz kayıtlara dair doğruluk sonuçları arasında düşük farkların bulunması, *k*-SLAM'in algılayıcı gürültüsüne olan dayanıklılığını ve yoğun yöntemlerin birçoğundan daha başarılı performans sergilediğini göstermektedir.

### 7.3.2. TUM RGB-D Benchmark Veri Kümesi Deneyleri

Bu veri kümesi daha önceki bölümlerde de açıklandığı üzere hem orta ölçekli (*fr1* kümesi) hem de geniş ortamlarda (*fr2* kümesi) kaydedilmiş görüntü kayıtlarına sahiptir. Daha önce gerçekleştirilen deneylerde de kullanılmış olan bu kayıtlara dair ayrıntılı bilgiler *fr1* ve *fr2* kümeleri için sırasıyla Çizelge 4.1 ve Çizelge 5.1'de listelenmiştir.

#### 7.3.2.1. *fr1* Görüntü Kayıtları

*fr1* kayıtları orta ölçekli bir ofis ortamında kaydedilmiş olup dokulu sahnelere sahiptirler, dolayısıyla anahtar nokta bakımından zenginlerdir (ortalama anahtar nokta sayıları için Bkz. Şekil 7.7). Sistemin *fr1* kayıtları üzerinde elde edilen doğruluk sonuçları Çizelge 7.4'te yer almaktadır. Burada *k*-SLAM sistemi şu yöntemlerle kıyaslanmıştır; Kerl ve diğerlerinin [51] geliştirdiği DVO SLAM sistemi, KinectFusion [54], Whelan ve diğerleri [57] tarafından geliştirilen Kintinuous (ICP+RGB-D kombinasyonu ve döngü kapama ile çizge optimizasyonu bileşenleriyle birlikte), Stückler ve Behnke [59] tarafından önerilen MRSSMap, Mur-Artal ve Tardos'un [47] geliştirdiği ORB-SLAM2, Whelan ve diğerlerinin [60] ElasticFusion sistemi, Endres ve diğerlerinin [44] önerdiği RGB-D SLAM ve bu sisteme etkili bir döngü kapama tespiti yönteminin entegre edildiği (Bkz. Bölüm 6) Genişletilmiş RGB-D SLAM [87]. Doğruluk karşılaştırması yapılırken, araştırmacılar tarafından yayınlanmış olan RMS-ATE sonuçları kullanılmıştır. Burada yer alan bir istisna, geliştiriciler tarafından *fr1* kümesi için sadece *fr1/desk*, *fr1/desk2* ve *fr1/room* sonuçlarına yer verilen ORB-SLAM2 sistemidir. Dolayısıyla

geliştirilen sistemle kıyaslama yapılabilmesi için, ORB-SLAM2 sistemi geliştiricileri tarafından sağlanan orijinal gerçekleştirimi kullanılarak veri kümesindeki diğer kayıtlar üzerinde çalıştırılmıştır. Ayrıca Bölüm 6'da tanıtılan Genişletilmiş RGB-D SLAM de tüm *fr1* kayıtları ile denenmiştir.

Çizelge 7.4. TUM RGB-D Benchmark veri kümesinde yer alan *fr1* kayıtları üzerinde mutlak yol hatası (RMS-ATE) karşılaştırması (m)

Sistem	<i>fr1/360</i>	<i>fr1/desk</i>	<i>fr1/desk2</i>	<i>fr1/floor</i>	<i>fr1/plant</i>	<i>fr1/room</i>	<i>fr1/rpy</i>	<i>fr1/teddy</i>	<i>fr1/xyz</i>
DVO SLAM	0.083	0.021	0.046	-	0.028	0.053	<b>0.020</b>	<b>0.034</b>	0.011
KinectFusion	0.913	0.057	0.420	-	0.598	0.313	0.133	0.154	0.026
Kintinous	-	0.037	0.071	-	0.047	0.075	0.028	-	0.017
MRSMap	0.069	0.043	0.049	x	0.026	0.069	0.027	0.039	0.013
ORB-SLAM2	0.180	<b>0.016</b>	<b>0.022</b>	x	<b>0.014</b>	0.047	<b>0.020</b>	0.047	<b>0.009</b>
ElasticFusion	0.108	0.020	0.048	x	0.022	0.068	0.025	0.083	0.011
RGB-D SLAM	0.079	0.026	0.043	0.035	0.091	0.087	0.026	0.076	0.014
Genişletilmiş RGB-D SLAM	0.075	0.022	0.034	0.032	0.068	0.054	0.022	0.060	0.014
<i>k</i> -SLAM	<b>0.053</b>	0.020	0.028	<b>0.027</b>	0.039	<b>0.046</b>	0.021	0.037	0.012

Çizelge 7.4'teki mutlak yol hatası sonuçları, *k*-SLAM'in en gelişmiş haritalama sistemleriyle başa baş ya da daha iyi performans sergilediğini göstermektedir. Geliştirilen sistemle elde edilen kayma hataları 1 cm - 5 cm aralığında değişmekte olup genellikle ORB-SLAM2 sistemiyle yarışmaktadır. *fr1/xyz*, *fr1/desk*, *fr1/desk2* ve *fr1/rpy* kayıtları kısıtlı bir alana odaklanılarak oluşturulmuşlar ve nispeten daha kısa yollar içermektedirler. Dolayısıyla sistem bu kayıtlar için daha düşük hata üretmektedir. *fr1/desk2* anahtar nokta açısından biraz daha düşük ortalamaya sahip olduğundan doğruluk oranı ters yönde etkilenmektedir. Diğer yandan, *fr1/xyz* daha düşük hızda ve küçük ölçekli kamera hareketleriyle kaydedildiğinden kayma hatası daha da düşmektedir. *fr1/floor* görüntü kaydında algılayıcı zemine çok yakın şekilde ve düşük hızda hareket ettirilmiştir. Bu durum, ortalama en yüksek anahtar nokta sayısına ulaşılmasına ve geliştirilen sistemin daha başarılı hareket tahmini yapmasına imkân vermektedir. Diğer kayıtların aksine, ORB-SLAM2, ElasticFusion ve MRSMap sistemlerinde *fr1/floor* için robot hareketi takibinde başarısızlıklar meydana gelmektedir. Çok kısa bir yol ve küçük bir döngü kapama içermesine rağmen, *fr1* kümesinde gözlemlenen en yüksek kayma hatası *fr1/360* görüntü kaydına aittir. Bu durumun temel sebebi görüntü kaydının düşük sayıda anahtar noktaya sahip olması ve dolayısıyla düşük kalitede dönüşümlerin

hesaplanmasıdır. *fr1/plant*, *fr1/room* ve *fr1/teddy* kayıtları nispeten daha uzun yollar ve küçük çaplı döngü kapamalar içermektedirler. Şekil 7.6, *k*-SLAM'in bu kayıtlar için döngü kapamaları etkin şekilde tespit edebildiğini ve gerçeğe yakın yol tahmini yapabildiğini göstermektedir. *fr1/room* için gözlemlenen az miktarda da olsa yüksek kayma hatası anahtar nokta sayısından kaynaklanmaktadır. Genel olarak *k*-SLAM sistemi, *fr1* kayıtlarında en başarılı ya da en başarılıya çok yakın performans sergilemektedir. Bu durumun tek istisnası, ORB-SLAM2 ile arada 2.5 cm RMS-ATE farkı bulunan *fr1/plant* görüntü kayıdır.

### 7.3.2.2. *fr2* Görüntü Kayıtları

*fr2* görüntü kayıtları geniş bir ortamda kaydedilmiş olup *fr1*'e göre daha zorlu durumlar içermektedirler. Bu kümede döngü kapama açısından zengin ve uzun yollara sahip kayıtlar mevcuttur. Bu kayıtlarla elde edilen kayma hatası sonuçları Çizelge 7.5, Çizelge 7.6 ve Çizelge 7.7'de listelenmiştir. Geliştirilen sistemin karşılaştırıldığı yöntemler; Kerl ve diğerleri [51] tarafından geliştirilen DVO SLAM, Whelan ve diğerlerinin [57] önerdiği Kintinuous (ICP+RGB-D kombinasyonu ve döngü kapama ile çizge optimizasyonu bileşenleriyle birlikte), Stückler ve Behnke'nin [59] geliştirdiği MRSSMap sistemi, Mur-Artal ve Tardos [47]'un ORB-SLAM2 sistemi, Whelan ve diğerlerinin [60] geliştirdiği ElasticFusion, Endres ve diğerleri tarafından geliştirilen RGB-D SLAM [44] ve Genişletilmiş RGB-D SLAM [87]'dir. Önceki kesimdeki gibi, ORB-SLAM2 sistemi geliştiriciler tarafından sağlanan orijinal gerçekleştirimi ile *fr2* kümesindeki kayıtlar için (yayında yer verilen *fr2/desk* kaydı dışında) denenmiştir.

DeneySEL sonuçlar geliştirilen sistemin *fr2* kayıtlarının birçoğunda en iyi sonuçları ürettiğini göstermektedir. *k*-SLAM sistemi, ORB-SLAM2 ve ElasticFusion gibi gelişmiş haritalama yöntemlerinin robot takibinde başarısız olarak haritalayamadığı çok zorlu ve geniş ölçekli görüntü kayıtlarında etkili performans sergileyebilmektedir. Bu sistemlerde hataya sebep olan temel faktörler, bazı kayıtlardaki kesintiler ve bazı sahnelerde büyük miktarda derinlik verisinin eksik olması gibi çerçeve çakıştırmalarını başarısız kılacak durumlar olabilir. Ayrıca geniş ölçekli haritalamada döngü kapamaların doğru tespit edilmesindeki zorluk da bir diğer temel etkidir. Tahmin edilen robot yolları (Bkz. Şekil 7.6), *k*-SLAM'in bu açıdan dayanıklılığını göstermektedir.



Çizelge 7.5. TUM RGB-D Benchmark veri kümesinde yer alan *fr2* kayıtları üzerinde mutlak yol hatası (RMS-ATE) karşılaştırması (m)

Sistem	fr2/360_hemisp.	fr2/coke	fr2/desk	fr2/dishes	fr2/flower.	fr2/flower_br.
DVO SLAM	-	-	0.017	-	-	-
Kintinuous	-	-	0.034	-	-	-
MRSMap	-	-	0.052	-	-	-
ORB-SLAM2	<b>0.113</b>	x	<b>0.009</b>	<b>0.058</b>	x	x
ElasticFusion	-	x	0.071	x	-	-
RGB-D SLAM	0.592	0.205	0.057	1.126	0.131	0.703
Genişletilmiş RGB-D SLAM	0.595	0.191	0.090	0.785	0.132	0.678
<i>k</i> -SLAM	0.460	<b>0.183</b>	0.068	0.731	<b>0.045</b>	<b>0.380</b>

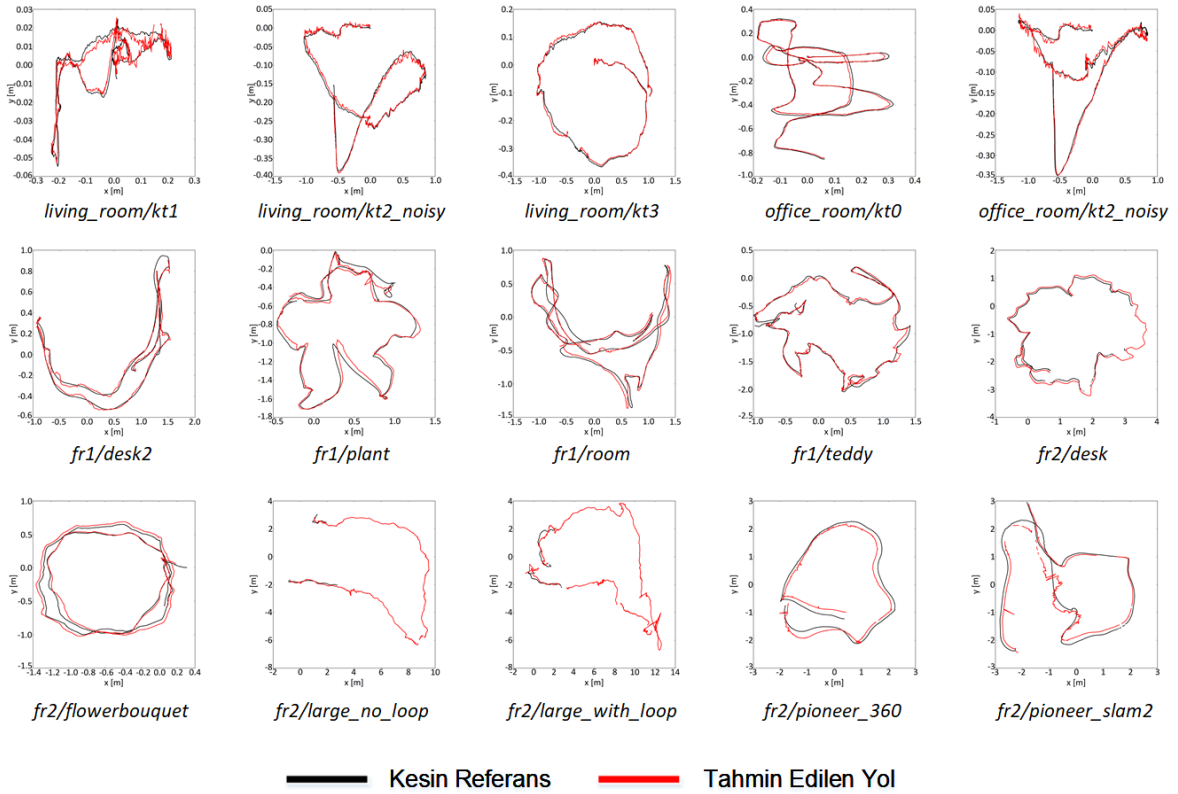
Çizelge 7.6. TUM RGB-D Benchmark veri kümesinde yer alan *fr2* kayıtları üzerinde mutlak yol hatası (RMS-ATE) karşılaştırması-2 (m)

Sistem	fr2/large_no_loop	fr2/large_with_loop	fr2/metallic_sphere	fr2/metallic_sphere2
DVO SLAM	-	-	-	-
Kintinuous	-	-	-	-
MRSMap	-	-	-	-
ORB-SLAM2	0.337	x	x	x
ElasticFusion	x	x	x	x
RGB-D SLAM	0.860	3.598	1.099	0.772
Genişletilmiş RGB-D SLAM	0.449	0.377	0.535	<b>0.703</b>
<i>k</i> -SLAM	<b>0.150</b>	<b>0.345</b>	<b>0.528</b>	0.748

Çizelge 7.7. TUM RGB-D Benchmark veri kümesinde yer alan *fr2* kayıtları üzerinde mutlak yol hatası (RMS-ATE) karşılaştırması-3 (m)

Sistem	fr2/pioneer_360	fr2/pioneer_slam	fr2/pioneer_slam2	fr2/pioneer_slam3
DVO SLAM	-	-	-	-
Kintinuous	-	-	-	-
MRSMap	-	-	-	-
ORB-SLAM2	x	x	x	x
ElasticFusion	x	x	x	x
RGB-D SLAM	0.213	0.367	0.381	0.511
Genişletilmiş RGB-D SLAM	0.213	<b>0.331</b>	0.392	<b>0.330</b>
<i>k</i> -SLAM	<b>0.149</b>	0.809	<b>0.183</b>	0.616

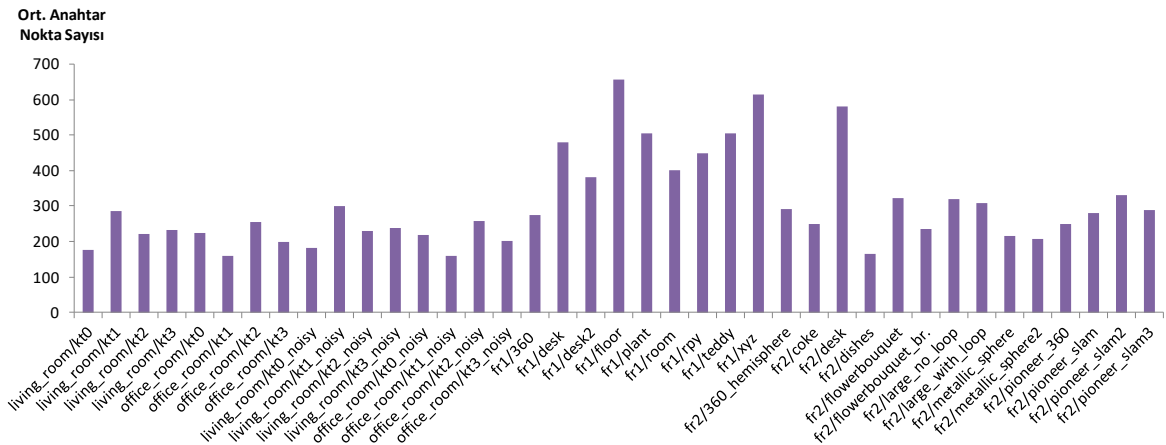
Diğer kayıtlara göre çok daha geniş alanları kapsayan ve uzun yollar içeren görüntü kayıtlarından *fr2/large\_no\_loop* için yanlış döngü kapama tespitleri yapılmaması daha kritiktir, çünkü bu kayıttaki herhangi bir döngü kapama mevcut değildir. Dolayısıyla bu kayıt için, uyarlanabilir eşikleme mekanizmasının modele aykırı adayları eleme performansı başarımı belirleyen önemli bir etkidir. *fr2/large\_with\_loop* görüntü kaydında ise mevcut olan tek ve çok geniş döngü kapamanın doğru tespit edilmesi büyük önem taşıdığından, bu denli büyük bir kayıt için geniş bir arama uzayında adayların doğru şekilde belirlenmesi kritiktir. Şekil 7.6'da görüldüğü üzere *k*-SLAM geniş ölçekli kayıtlarda etkili performanslar sergilemiştir.



Şekil 7.6. Örnek görüntü kayıtları için, tahmin edilen yol ve kesin referansın x-y düzlemine izdüşümü

Algılayıcının robot üzerine monte edilerek yalnızca yatay düzlemde yapılan yer değiştirmelerle kaydedilmiş olan *fr2/pioneer\_360*, *fr2/pioneer\_slam*, *fr2/pioneer\_slam2* ve *fr2/pioneer\_slam3* görüntü kayıtlarında stabil robot hareketleri dönüşüm hesaplamasını kolaylaştırır da görüntüdeki kesilmeler ve derinlik verisindeki eksiklikler zorlaştırıcı etkenlerdir. *fr2/pioneer\_360* kaydı

endüstriyel salonun ortasında bir döngü içerirken, diğer üç kayıt daha fazla sayıda döngü kapamaya sahiptir. Elde edilen sonuçlar, *fr2/pioneer\_360* ve *fr2/pioneer\_slam2* için sırasıyla yaklaşık 15 cm ve 18 cm RMS-ATE değerleriyle daha yüksek başarıma işaret etmektedir. *fr2/pioneer\_slam* ve *fr2/pioneer\_slam3* kayıtları için ise daha fazla kayma hatası gözlemlenmiştir. Bu kayıtlarda Genişletilmiş RGB-D SLAM, genel görüntü niteliklerinin yanında yerel niteliklerden de faydalanması sayesinde daha başarılı performans göstermiştir.



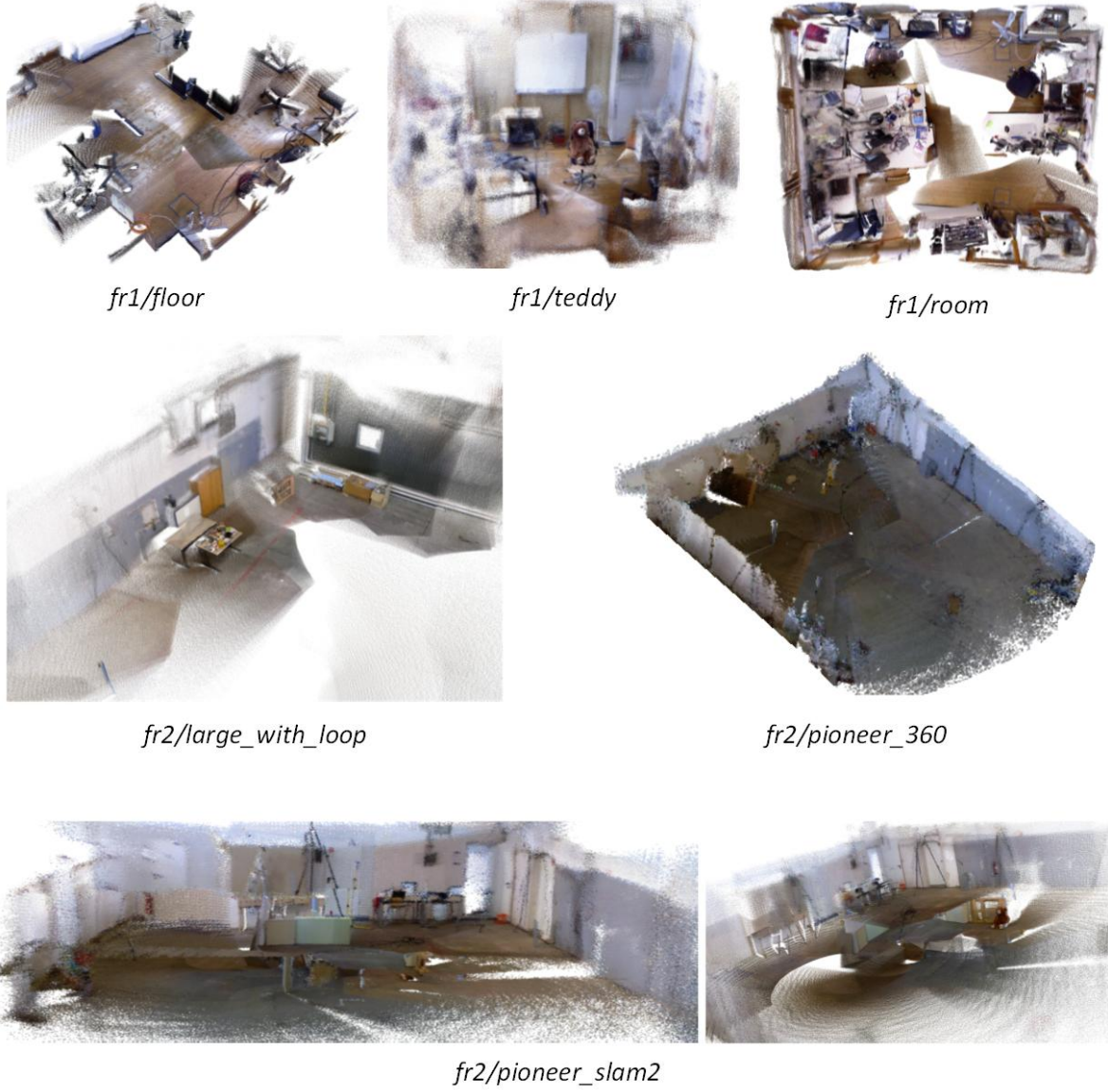
Şekil 7.7. Her bir kayıt için elde edilen çerçeve başına ortalama anahtar nokta sayıları

*fr2/desk* kaydı ortalamada kümedeki diğer tüm kayıtlardan çok daha fazla sayıda anahtar nokta içermektedir (Bkz. Şekil 7.7). Bu durum, yüksek sayıda modele uygun eşleşme içeren daha başarılı dönüşümlerin tahmin edilebilmesine imkân vermektedir. Ayrıca geçersiz derinlik verisi oranı bu kayıt için nispeten daha düşüktür. Dolayısıyla bu faktörler algılayıcı hareketinin daha kolay takip edilebilmesini ve dolayısıyla odometri tahmini performansının sistem başarımını belirleyen temel etken olmasını sağlamaktadırlar. Bu kayıt için ORB-SLAM2 daha başarılı performans sergilemektedir. *fr2/flowerbouquet* ve *fr2/flowerbouquet\_br.* görüntü kayıtlarındaki robot yolları şekil bakımından *fr2/desk*'e benzemektedirler. *fr2/flowerbouquet* için yol uzunluğu ve algılayıcı hızının *fr2/desk*'in yaklaşık yarısı kadar olması, sahip olduğu sahnelerdeki düşük dokuya rağmen kayma hatasındaki artışı yavaşlatan faktörlerdir. *fr2/flowerbouquet\_br.* kaydında ise hem algılayıcının daha hızlı hareket ettirilmesi hem de arka planın bir kartonla

kapatılarak dokunun azaltılması hatayı arttırmaktadır. *fr2/coke* kaydı nispeten daha az zorlu bir kayıt olmasına rağmen anahtar noktaların çerçevelerin kısıtlı bölgelerinde kümelenmiş olması dönüşüm kalitesini düşürebilmektedir. Ortalamada en düşük dokuya sahip kayıtlardan *fr2/dishes*, *fr2/metallic\_sphere* ve *fr2/metallic\_sphere2* nispeten kısa robot yolları içermelerine rağmen, düşük sayıda ve çerçeve üzerinde dağılmamış anahtar noktalar sebebiyle bu kayıtlar için benzer kayıtlardan daha yüksek hata gözlemlenmektedir. *fr2/dishes* ve *fr2/360\_hemisphere* kayıtlarında ORB-SLAM2 daha etkili performans sergilemiştir.

Sonuç olarak, gözlemlendiği üzere *k*-SLAM sistemi diğer gelişmiş haritalama sistemlerinin çoğunlukla başarısız olduğu zorlu haritalama koşullarında etkili performanslar sergilemektedir. Sistemdeki nitelik tabanlı hareket tahmini bileşeni, güvenilir dönüşüm hesaplaması için yoğun yöntemlerdeki gibi art arda süregelen çerçeve mantığına ihtiyaç duymamaktadır. Dolayısıyla görüntüdeki kesilmeler hareket tahmini performansını önemli derecede etkilememektedir. Diğer yandan, geniş ölçekli haritalamada döngü kapama tespiti önemli bir zorluktur. Ortam genişledikçe arama uzayı da sürekli genişlediğinden, doğru döngü kapama tespiti daha zor ve zaman alıcı hale gelmektedir. Bu büyük problemten dolayı birçok gelişmiş SLAM sistemi geniş ortamlarda başarılı olamamaktadır. Autocorrelogram indeksleme tabanlı döngü kapama tespiti yaklaşımı, sistemin geniş ölçekli ortamlarda da etkili ve verimli şekilde doğru döngü kapama adaylarını bulabilmesini sağlamaktadır. Görsel olarak benzer bir döngü kapama adayı doğru aday olmayabilir, çünkü haritanın tutarlılığı döngüyü kapayan çerçeveler arasında hesaplanan dönüşümün kalitesine bağlıdır. Hesaplanan dönüşüm doğrulanmasına rağmen düşük sayıda modele uygun değer içeriyor ya da eşleşen anahtar noktalar çerçeveler üzerinde kısıtlı bir bölgede yer alıyorsa dönüşüm kalitesi olumsuz etkilenebilir ve hata artabilir. Dolayısıyla tespit edilen döngü kapamaların artması sistem başarımının artması manasına gelmemektedir. Önemli olan, yüksek kalitede dönüşümler hesaplanabilecek doğru döngü kapamaların tespit edilebilmesidir. Sistemdeki autocorrelogram benzerliği tabanlı uyarlanabilir eşikleme mekanizmasının amacı, modele aykırı (mevcut çerçeveye benzemeyen) adayların yanında bu tarz daha düşük benzerliğe sahip adayların da elenmesidir. Elde edilen deneysel sonuçlar, uyarlanabilir eşiklemenin bu açıdan etkisini

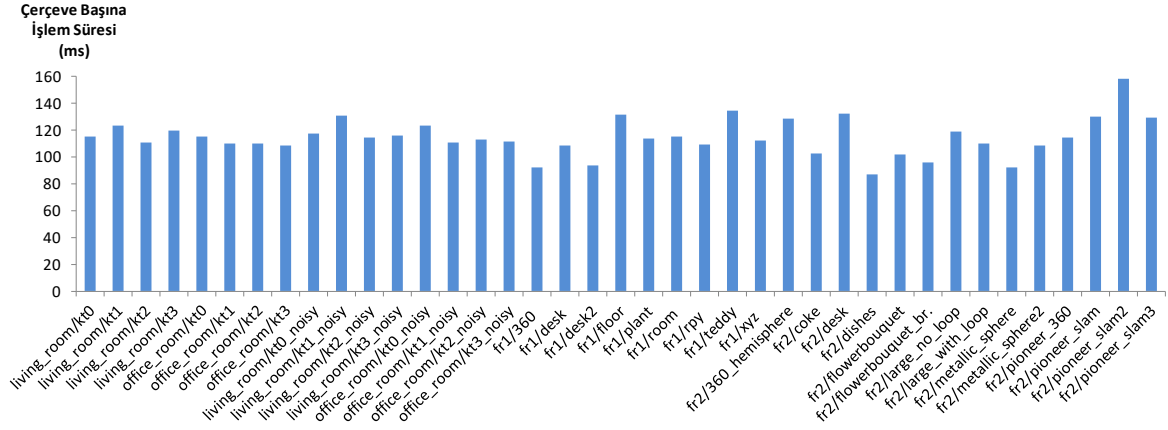
göstermektedir. Şekil 7.8'de, farklı görüntü kayıtları için  $k$ -SLAM sisteminin ürettiği yoğun nokta bulutu yapıları görülmektedir.



Şekil 7.8. Örnek kayıtlar için sistem tarafından üretilen nokta bulutu gösterimleri

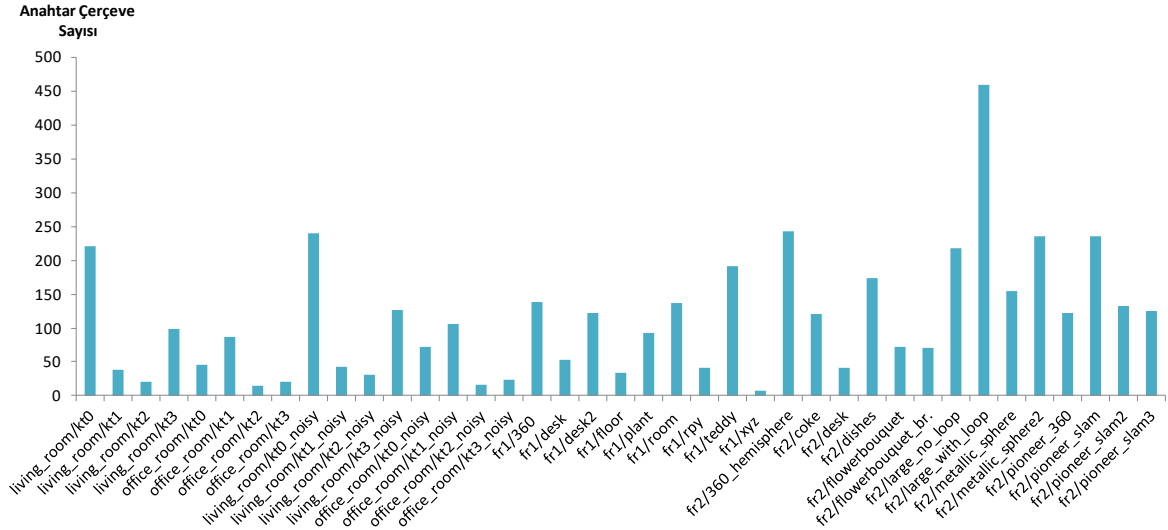
### 7.3.3. İşlemsel Performans

Şekil 7.9 ve Şekil 7.10'da sırasıyla,  $k$ -SLAM ile veri kümelerindeki tüm kayıtlar için elde edilen ortalama işlem süreleri ve anahtar çerçeve sayıları yer almaktadır.



Şekil 7.9. Her bir kayıt için elde edilen çerçeve başına işlem süresi (ms)

İşlem süresi sonuçları, sitemin çalışma hızının yaklaşık olarak 9 Hz olduğunu göstermektedir. *k*-SLAM'in diğer tüm haritalama sistemlerine kıyasla en büyük avantajı, haritalanan ortamın küçük, orta ölçekli ya da geniş olmasına bakılmaksızın kararlı şekilde işlemsel performans sergilemesidir. Sistem, haritalanan ortamın büyüklüğünden bağımsız olarak döngü kapamaları oldukça hızlı şekilde tespit edebilmektedir. Harita büyüdükçe daha fazla gözlem verisi içerisinde arama yapılması gerektiğinden döngü kapama tespiti maliyetinin artması, dolayısıyla işlem süresinin uzaması beklenir. Ancak geliştirilen sistem döngü kapamaları anahtar çerçeve autocorrelogramlarından oluşan bir ağaç yapısı içerisinde aradığından, ortam geniş ve robot yolu uzun olsa da arama işlemi çok düşük bir maliyet getirmektedir. En geniş alanı kapsayan ve en uzun iki robot yolundan birisine sahip olan *fr2/large\_with\_loop* görüntü kaydı 459 adet anahtar çerçeve içermektedir (Bkz. Şekil 7.10). 459 autocorrelogramdan oluşan bir öncelik araması *k*-ortalamalar ağacında ihmal edilebilecek kadar düşük bir maliyetle arama yapılabilir çünkü autocorrelogram karşılaştırmaları çok düşük düzeyde işlemsel yük getirmektedir. Bunun yanı sıra, uyarlanabilir eşikleme sadece benzerlik sonuçlarını karşılaştıran verimli bir mekanizmadır. Dolayısıyla, herhangi bir çerçeve için gerçekleştirilen döngü kapama aday seçimi işleminin tamamı 1 milisaniyeden daha kısa sürmektedir. İşlemsel sürenin büyük kısmı nitelik çıkarımı (~40 ms) ve hareket tahmini (~45 ms) işlemleri tarafından harcanmaktadır.



Şekil 7.10. Her bir kayıt için elde edilen anahtar çerçeve sayısı

Ortalama anahtar nokta sayısı, işlem süresi üzerinde etkili olan bir diğer faktördür. Örneğin *fr1/floor*, *fr1/xyz* ve *fr1/desk* gibi yüksek dokulu kayıtlar daha fazla işlemsel çaba gerektirmektedir, çünkü hareket tahmini aşamasındaki RANSAC yinelemelerinde daha fazla sayıda anahtar nokta ele alınmaktadır. Ayrıca, odometri tahmini için kullanılan direkt önceki çerçevelerin sayısı bir diğer önemli etkindir. Mevcut çerçeve ile direkt önceki çerçevelerin her biri arasında hareket tahmini gerçekleştirildiğinden, daha fazla sayıda önceki çerçeve daha fazla işlem süresi anlamına gelmektedir (örneğin *fr2/pioneer\_slam*, *fr2/pioneer\_slam2* ve *fr2/pioneer\_slam3*).

Çizelge 7.8'de *k*-SLAM'in *fr2* kümesindeki çalışma hızı diğer nitelik tabanlı SLAM sistemleri ile kıyaslanmıştır. Karşılaştırma yapılan sistemler; RGB-D SLAM [44], Genişletilmiş RGB-D SLAM [87] ve ORB-SLAM2 [47]'dir. Çizelgedeki sonuçlar, *k*-SLAM'in hem RGB-D SLAM'den hem de genişletilmiş versiyonundan çok daha hızlı çalıştığını göstermektedir. ORB-SLAM2, *fr2/large\_no\_loop* kaydında daha düşük maliyetle çalışmaktadır. ORB-SLAM2 için daha düşük maliyeti sağlayan temel etken, anahtar çerçevelerden oluşan bir çizge kullanılmasıdır. Böyle bir çizge, her bir çerçeve için çizgede bir düğüm oluşturan *k*-SLAM'e göre çok daha seyrek bir yapıda olduğundan maliyeti düşürmektedir. Bu duruma rağmen, yüksek dokudan daha fazla etkilenen ORB-SLAM2 sistemi *fr2/desk* için daha fazla işlem maliyetiyle çalışmaktadır.

Çizelge 7.8. Diğer nitelik tabanlı SLAM sistemleri ile işlemsel performans karşılaştırması (ms)

Görüntü Kaydı	<i>k</i> -SLAM	RGB-D SLAM	Genişletilmiş RGB-D SLAM	ORB-SLAM2
fr2/360_hemisphere	128.9	154.1	181.4	70.0
fr2/coke	102.9	167.3	184.7	-
fr2/desk	132.4	348.6	428.3	137.8
fr2/dishes	87.0	135.4	147.6	75.0
fr2/flowerbouquet	102.2	202.6	228.7	-
fr2/flowerbouquet_br.	95.7	149.8	174.3	-
fr2/large_no_loop	119.0	143.9	166.3	89.4
fr2/large_with_loop	110.1	156.6	185.8	-
fr2/metallic_sphere	92.5	155.7	172.4	-
fr2/metallic_sphere2	108.4	136.4	154.2	-
fr2/pioneer_360	114.6	139.8	158.5	-
fr2/pioneer_slam	130.0	155.4	173.4	-
fr2/pioneer_slam2	158.1	198.9	222.6	-
fr2/pioneer_slam3	129.4	153.4	171.2	-

#### 7.4. Değerlendirme

Bu çalışmada, RGB-D algılayıcı kullanan bir 3B SLAM sistemi geliştirilmiştir. Sistem görsel nitelik çıkarımı ve eşleştirmesini kullanmakta ve çerçeveler arasındaki hareketi RANSAC algoritmasını uygulayarak hesaplamaktadır. Döngü kapama tespiti için, sıkı bir genel görüntü tanımlayıcısı olan autocorrelogram yapısı kullanılarak etkili bir mekanizma geliştirilmiştir. Döngü kapamalar, mevcut ve önceki algılayıcı gözlemleri arasındaki görsel ilişkilerin, görüntüler arasındaki autocorrelogram benzerliklerinden faydalanılarak bulunması yoluyla tespit edilmektedir. Anahtar çerçeve autocorrelogramları bir öncelik araması *k*-ortalamlar ağacı içerisinde sıradüzensel şekilde indekslenmekte ve döngü kapama araması ağacın özyineli olarak dolaşılması yoluyla gerçekleştirilmektedir. Uyarlanabilir eşikleme yaklaşımı kullanılarak modele aykırı adaylar dinamik şekilde elendikten sonra döngü kapama adayları belirlenmektedir. Çizge optimize edilerek robot yolunun belirlenmesiyle ortamın haritası oluşturulmaktadır.

Sistemin performansı, popüler veri kümelerinin içerdiği çok sayıda ve zorlu şartlara sahip görüntü kayıtlarının kullanıldığı ayrıntılı deneylerle analiz edilmiştir. Deneysel sonuçlar, sistemin kapalı ortamları etkili şekilde haritalayabildiğini ve gelişmiş RGB-D SLAM sistemlerinin çoğunluğundan daha başarılı olduğunu göstermektedir. Özellikle yakın zamanda geliştirilmiş önemli haritalama



yöntemlerinin ortamdaki zorlu şartlardan dolayı başarısız olduđu geniş ölçekli görüntü kayıtları için ümit verici sonuçlar elde edilmiştir. Sistem, geniş ölçekli haritalarda dahi CPU üzerinde gerçek zamanlı olarak çalışabilmektedir. Geliştirilen döngü kapama tespiti yöntemi, zorlu ortam şartlarına karşı dayanıklılık sağlamanın yanında geniş ölçekli haritalama için yeterince hızlı çalışabilmektedir.

## 8. SONUÇLAR

Bu tez kapsamında, Eş Zamanlı Konum Belirleme ve Haritalama (SLAM) alanındaki temel problemler üzerinde çalışılmıştır. RGB-D algılayıcı kullanılarak kapalı ortamların haritalarının oluşturulması alanında çeşitli çalışmalar gerçekleştirilerek yeni yöntemler geliştirilmiştir.

Tez çalışmasının ilk aşamasında, algılayıcıdan elde edilen veri çerçevelerinin karşılaştırılması yoluyla robot hareketinin tahmin edilmesi problemine odaklanılmıştır. Bu problemin çözümüne yönelik iki temel yaklaşımdan birisi olan nitelik tabanlı çerçeve karşılaştırma tekniği üzerinde bir analiz çalışması gerçekleştirilmiştir. Bu teknikte veri çerçeveleri üzerindeki belirgin anahtar noktalar nitelik bulucularla tespit edilmekte ve bu noktaları tanımlayan vektörler nitelik tanımlayıcılar tarafından oluşturulmaktadır. Aralarında hareket tahmini yapılacak olan çerçeveler arasında bu anahtar noktalar eşleştirilmekte ve eşleşmeler üzerinden dönüşüm tahmini gerçekleştirilmektedir. Bu bağlamda bir analiz gerçekleştirilerek, hareket tahmininin başarımı açısından kritik öneme sahip olan nitelik bulucular ve nitelik tanımlayıcıların haritalama performansı üzerindeki etkileri incelenmiştir. Nitelik tabanlı hareket tahmini gerçekleştiren bir RGB-D SLAM sistemi üzerinde, değişik nitelik bulucu ve nitelik tanımlayıcı kombinasyonlarının kullanılması yoluyla elde edilen sonuçlar hız ve doğruluk açısından incelenmiştir. Elde edilen deneysel sonuçlar ışığında, CenSurE nitelik bulucusunun FREAK nitelik tanımlayıcısı ile birlikte kullanılmasının hız ve harita doğruluğu bakımından en etkili kombinasyon olduğu sonucuna varılmıştır. Bu ikili, çerçeve başına 165.7 ms süre harcamakta ve ortalama 3.79 cm mutlak yol hatası üretmektedir. Ayrıca kullanılan nitelik bulucu ve tanımlayıcı kombinasyonunun sistem performansını direkt olarak etkilediği gözlemlenmiştir. Bu analiz çalışması neticesinde, sonraki çalışmalarda çerçeveler arasındaki robot hareketinin tahmini için CenSurE nitelik bulucusu ile FREAK nitelik tanımlayıcısı birlikte kullanılmıştır.

Tez çalışmasının bir sonraki aşamasında, döngü kapama tespiti olarak bilinen, robotun ortamda daha önce geçtiği bir bölgeye tekrar uğradığının anlaşılabilmesi problemine odaklanılmıştır. Oluşturulan ortam haritasının tutarlı olabilmesi için döngü kapamaların başarılı şekilde tespit edilmesi gereklidir. Ancak ortam genişledikçe doğru tespit zorlaşması ve işlemsel yükün gittikçe artması, özellikle geniş ortamlar için bu problemin çözümünü hayli güçleştirmektedir. Bu bağlamda,

SLAM sistemlerinde kullanılabilir bir döngü kapama tespiti yöntemi geliştirilmiştir. Algılayıcıdan gelen renkli çerçevelerdeki genel ve yerel görüntü niteliklerini birlikte kullanan yöntem, iki-geçişli bir algoritma kullanarak anahtar çerçeveler arasından döngü kapama adaylarını belirlemektedir. Genel görüntü nitelikleri olarak histogram ve autocorrelogram yapılarını kullanmakta olan yöntem, yerel görüntü niteliği olarak ise odometri tahmini aşamasında zaten üretilmiş olan anahtar nokta tanımlayıcılarından faydalanmaktadır. Yöntemin ilk adımında, genel niteliklerin benzerliğine göre hızlı şekilde ve düşük maliyetle arama çerçevesine benzer bir grup anahtar çerçeve elde edilmektedir. Sonraki aşamada ise bu gruptaki her bir anahtar çerçeve ile arama çerçevesi arasında anahtar nokta eşleştirmesi gerçekleştirilerek, eşleşme sayılarına göre nihai döngü kapama adayları seçilmektedir. Geliştirilen döngü kapama tespiti yöntemi, küçük ölçekli ortamlarda nispeten başarılı sonuçlar üreten ancak döngü kapama tespitini rastgelelik içeren bir yaklaşımla ele aldığı için geniş ortamlarda aynı başarıyı sergileyemeyen bir RGB-D SLAM sistemine entegre edilmiştir. Bu kapsamda, döngü kapama tespiti yönteminin entegrasyonu ile birlikte geniş ölçekli ortam haritalaması açısından performans değerlendirmesi yapılmıştır. SLAM sistemlerinin başarımlarını ölçümünde yaygın olarak kullanılan TUM RGB-D Benchmark veri kümesi üzerinde gerçekleştirilen kapsamlı deneyler sonucunda, geliştirilen yöntemin döngü kapama adaylarını etkili şekilde tespit edebildiği gözlemlenmiştir. Özellikle geniş ölçekli haritalama senaryolarında önemli başarı sergileyen yöntemin, döngü kapama adaylarını rastgele seçen orijinal RGB-D SLAM sistemi ile elde edilen mutlak yol hatasını yaklaşık %45 oranında azalttığı görülmüştür. Bu başarı oranı, 128 renk seviyesine nicelendirilen görüntü üzerinde 2 piksel uzaklığına göre hesaplanan RGB autocorrelogramların Relative uzaklık metriğine göre karşılaştırılması ve ardından nitelik eşleştirmesi yapılması yoluyla elde edilmiştir. Yerel ve genel görüntü niteliklerinin birlikte kullanıldığı teknik %33 civarına bir ekstra işlem maliyeti getirmektedir. Bununla birlikte, yalnızca genel niteliklerin kullanımıyla çok az bir ekstra maliyetle nispeten daha düşük başarımların artırımı (~%37) da sağlanabilmektedir. Ayrıca bu çalışma neticesinde, autocorrelogram veri yapısının SLAM probleminde döngü kapama tespiti için etkin şekilde kullanılabilen bir alternatif olduğu gösterilmiştir.

Sonraki çalışmada, geliştirilmiş olan döngü kapama tespiti yönteminin daha hızlı ve verimli çalışabilmesi için yeni bileşenler geliştirilerek yöntem genişletilmiştir. Geniş ölçekli ortamların haritalanmasında gittikçe büyüyen arama uzayı önemli bir problem haline geldiğinden, literatürde mevcut olan SLAM sistemlerinin büyük çoğunluğu geniş ortamlarda gerçek zamanlı çalışma yeteneğine sahip değildir. Bu kapsamda, geliştirilen döngü kapama tespiti yöntemine sıradüzensel kümeleme ve uyarlanabilir eşikleme adında iki yeni bileşen eklenmiştir. Sıradüzensel kümeleme bileşeni çerçeveler arasında nitelik eşleştirmesi için, yöntemin önceki versiyonundaki gibi *brute-force* yaklaşımın uygulanması yerine ağaç veri yapısı tabanlı bir yaklaşık en yakın komşu algoritmasını kullanmaktadır. Anahtar çerçevelere ait nitelik tanımlayıcıları için sıradüzensel kümeleme ağaçları oluşturulmakta ve haritalama süresince eşleşme işlemi bu ağaçlarda özyineli arama yapılarak gerçekleştirilmektedir. Bu sayede *brute-force* yönteme yakın eşleşme başarısı elde edilirken, eşleşme işleminin hızı yaklaşık iki katına çıkmaktadır. Bir diğer yeni bileşen olan uyarlanabilir eşikleme, genel tanımlayıcı benzerliğine göre elde edilen başlangıç grubundaki mevcut çerçeveye daha az benzer anahtar çerçevelerin dinamik şekilde elenmesini sağlamaktadır. Bu sayede hem nitelik eşleştirme aşamasına daha az çerçeve ile devam edilerek maliyet düşürülmekte, hem de modele aykırı adaylar elenerek yanlış döngü kapama tespiti olasılığı daha da azaltılmaktadır. Uyarlanabilir eşikleme bileşeninin kullanımıyla döngü kapama tespiti maliyeti, yalnızca sıradüzensel kümeleme entegrasyonuna oranla yaklaşık %25 oranında düşürülmüştür. Dolayısıyla, sisteme eklenen iki yeni bileşenin birlikte kullanımı, döngü kapama tespitinin maliyetini önceki versiyona göre %64 civarında azaltmaktadır. Gerçekleştirilen kapsamlı deneylerde, harita doğruluğu açısından sistem performansının korunduğu gözlemlenmiş olup, mutlak yol hatası orijinal RGB-D SLAM sistemine göre yaklaşık %46 oranında düşürülmüştür. Bu başarı artışı için; ilk adımda 128 renk seviyesine nicelendirilmiş görüntüde 2 piksel uzaklığı kullanılarak hesaplanan Gri tonlamalı autocorrelogramlar Relative metriğine göre karşılaştırılarak başlangıç aday grubu elde edilmiştir. Sonrasında ise uyarlanabilir eşikleme uygulanarak bu gruptan eleme yapılmış ve sıradüzensel kümeleme ile nitelik eşleştirmesi gerçekleştirilerek döngü kapama adayları elde edilmiştir. Genişletilmiş döngü kapama tespiti yönteminin genel olarak getirdiği ekstra maliyet CPU üzerinde çerçeve başına yaklaşık 30 ms'dir. Sonuç olarak, genişletilmiş döngü kapama tespiti yöntemi

özellikle geniş ölçekli ortamlarda döngü kapama adaylarını başarılı şekilde tespit edebilmekte ve genişleyen arama uzayı problemine rağmen hızlı ve düşük maliyetle çalışabilmektedir. Nitelik tabanlı SLAM sistemlerine entegrasyonu oldukça kolay olan bu yöntem, geniş ortamların etkin şekilde haritalanabilmesi için kullanılabilir önemli bir seçenektir.

Tez çalışmasının sonraki aşamasında, önceki çalışmalardan elde edilen tecrübeler yardımıyla RGB-D algılayıcı tabanlı bir 3B SLAM sistemi geliştirilmiştir. Küçük çaplı, orta ölçekli ya da geniş ortamları haritalayabilen bu haritalama sistemi *k*-SLAM olarak adlandırılmıştır. Sistem harita çıkarımı için çizge tabanlı SLAM yaklaşımını kullanmaktadır. Hareket tahmini için çerçeveler arasındaki dönüşüm anahtar nokta eşleşmeleri üzerinden hesaplanmaktadır. Dolayısıyla *k*-SLAM nitelik tabanlı bir sistemdir. Döngü kapama tespiti, literatürde yer alan diğer gelişmiş SLAM sistemlerinden farklı olarak, genel görüntü niteliklerinin etkili bir veri yapısı içerisinde indekslenmesi yoluyla gerçekleştirilmektedir. Anahtar Çerçeve Autocorrelogram Veritabanı adı verilen bu yapıda, görüntülere ait gri tonlamalı autocorrelogramlar bir öncelik araması *k*-ortalamlar ağacı içerisinde sıradüzensel olarak indekslenmektedir. Her bir döngü kapama aramasında ilk olarak bu veri tabanından belirli sayıda döngü kapama adayı seçilmekte, sonrasında ise daha önce geliştirilmiş olan uyarlanabilir eşikleme mekanizması kullanılarak modele aykırı adaylar gruptan çıkarılmaktadır. Algılayıcıdan elde edilen her yeni çerçeve için çizgeye bir düğüm eklenerek, belirli sayıda önceki çerçeve ve döngü kapama adayı olarak seçilen anahtar çerçeveler ile dönüşüm hesaplaması gerçekleştirilmektedir. Doğrulanmış dönüşümler için çizgede ilgili düğümler arasında kenar oluşturulmakta ve tüm çerçeveler işlendikten sonra optimizasyon gerçekleştirilmektedir. Bu şekilde robotun yapmış olduğu yol elde edilerek ortamın haritası oluşturulmaktadır. *k*-SLAM sistemi, bu alanda yaygın olarak kullanılan veri kümeleri üzerinde çok sayıda görüntü kaydı ve çeşitli zorluklara sahip sahnelerde kapsamlı olarak test edilmiştir. Elde edilen sonuçlara göre *k*-SLAM, literatürde yer alan birçok gelişmiş haritalama sisteminden daha etkili performans sergilemektedir. Nispeten küçük ortamlarda 5 cm'den daha düşük ortalama kayma hatasıyla haritalama yapabilen *k*-SLAM, geniş ortamlarda ise daha yüksek hata içeren ancak ümit vadeden sonuçlar üretmiştir. Ortamın küçük, orta ölçekli ya da geniş olmasından bağımsız şekilde yakın maliyetlerle çalışabilen sistem, CPU

üzerinde ortalama olarak yaklaşık 9 Hz hızında haritalama yapabilmektedir. Bu özelliğiyle diğer SLAM sistemlerinden ayrılan  $k$ -SLAM için, ortam genişliğinden pek etkilenmeden döngü kapama adaylarını çok düşük maliyetle tespit edebilmesi önemli rol oynamaktadır. Bu faktör, günümüzdeki birçok gelişmiş SLAM sisteminin harita çıkarımında başarısız olmasına sebep olmaktadır.  $k$ -SLAM, geniş ortamlar dahil CPU üzerinde gerçek zamanlı şekilde haritalama yapabilmekte ve zorlu ortam şartlarına rağmen tutarlı haritalar oluşturabilmektedir.

Bu tez çalışmasında genel olarak döngü kapama tespiti problemi üzerine yoğunlaşmış olup, genel ve yerel görüntü niteliklerini kullanan çözümler geliştirilmiştir. İleriki çalışmalarda, farklı türlerdeki görüntü niteliklerinin sisteme adapte edilmesi hedeflenmektedir. Mevcut sistemde kullanılmakta olan autocorrelogram yapısının yanında, derin (*deep*) niteliklerin de yonteme entegre edilmesiyle harita oluşturma başarımının artırılması amaçlanmaktadır. Bunun yanı sıra, döngü kapama adayları tespit edilirken görsel benzerliğin yanında konum verisinin de ele alınması hedeflenen bir diğer aşamadır. Ayrıca odometri tahmininde daha başarılı sonuçlar üretebilen yoğun bir çerçeve çakıştırma yönteminin, uzak çerçeveler arasında daha iyi sonuç veren nitelik tabanlı hareket tahmini bileşeni ile birlikte kullanımıyla hibrid bir yaklaşım uygulanarak robot yolunun daha başarılı tahmin edilebileceği düşünülmektedir. Bir diğer hedef ise sistemin GPU üzerinde çalışabilen bir versiyonunun üretilmesi yoluyla işlem hızının daha da yükseltilmesidir.

## KAYNAKLAR

- [1] Thrun, S., Leonard, J. J., Simultaneous Localization and Mapping, *Springer Handbook of Robotics*, 871–889, **2008**.
- [2] Autonomous Navigation for Flying Robots Lecture Notes, <https://vision.in.tum.de/teaching/ss2015/autonavx> (Mart, **2018**).
- [3] Thrun, S., Burgard, W., Fox, D., *Probabilistic Robotics*, MIT press, **2005**.
- [4] Grisetti, G., Kummerle, R., Stachniss, C., Burgard, W., A tutorial on graph-based SLAM, *IEEE Intelligent Transportation Systems Magazine*, 2, 31–43, **2010**.
- [5] Scaramuzza, D., Fraundorfer, F., Visual Odometry [Tutorial], *IEEE Robotics & Automation Magazine*, 18, 80–92, **2011**.
- [6] Fraundorfer, F., Scaramuzza, D., Visual odometry: Part ii: Matching, robustness, optimization, and applications, *IEEE Robotics & Automation Magazine*, 19, 78–90, **2012**.
- [7] Lowry, S. *et al.*, Visual place recognition: A survey, *IEEE Transactions on Robotics*, 32, 1–19, **2016**.
- [8] Autonomous Mobile Robots Lecture Notes, [http://www.asl.ethz.ch/education/lectures/autonomous\\_mobile\\_robots/spring-2017.html](http://www.asl.ethz.ch/education/lectures/autonomous_mobile_robots/spring-2017.html) (Mart, **2018**).
- [9] Bailey, T., Durrant-Whyte, H., Simultaneous localization and mapping (SLAM): Part I, *IEEE Robotics and Automation Magazine*, 13, 108–117, **2006**.
- [10] Robot Mapping Lecture Notes, <http://ais.informatik.uni-freiburg.de/teaching/ws17/mapping/> (Nisan, **2018**).
- [11] Bailey, T., Durrant-Whyte, H., Simultaneous localisation and mapping (slam) part 2: State of the art, *IEEE Robotics & Automation Magazine*, 13, 108–117, **2006**.
- [12] Thrun, S., Burgard, W., Fox, D., A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping, *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, 1, 321–328, **2000**.
- [13] Triebel, R., Burgard, W., Improving Simultaneous Mapping and Localization in 3D Using Global Constraints, *Proceeding of the National Conference on Artificial Intelligence (AAAI)*, 1330–1335, **2005**.
- [14] Yap, T. N., Shelton, C. R., SLAM in large indoor environments with low-cost, noisy, and sparse sonars, *Proceedings - IEEE International Conference on Robotics and Automation*, 1395–1401, **2009**.
- [15] Konolige, K., Agrawal, M., FrameSLAM: From bundle adjustment to real-time visual mapping, *IEEE Transactions on Robotics*, 24, 1066–1077, **2008**.
- [16] Clemente, L. A., Davison, A. J., Reid, I. D., Neira, J., Tardos, J. D., Mapping Large Loops with a Single Hand-Held Camera, *2007 Robotics: Science and Systems Conference*, 297–304, **2007**.

- [17] Davison, A. J., Real-Time Simultaneous Localisation and Mapping with a Single Camera, *International Conference on Computer Vision*, 1403–1410, **2003**.
- [18] Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B., FastSLAM: A factored solution to the simultaneous localization and mapping problem, *Proc. of 8th National Conference on Artificial Intelligence/14th Conference on Innovative Applications of Artificial Intelligence*, 68, 593–598, **2002**.
- [19] Montemerlo, M., Thrun, S., Roller, D., Wegbreit, B., FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges, *IJCAI International Joint Conference on Artificial Intelligence*, 1151–1156, **2003**.
- [20] Lu, F., Milios, E., Globally consistent range scan alignment for environment mapping, *Autonomous robots*, 4, 333–349, **1997**.
- [21] Grisetti, G., Kummerle, R., Stachniss, C., Burgard, W., A Tutorial on Graph-Based SLAM, *IEEE Intelligent Transportation Systems Magazine*, 2, 31–43, **2010**.
- [22] Cadena, C. *et al.*, Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age, *IEEE Transactions on Robotics*, 32, 1309–1332, **2016**.
- [23] Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D., RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments, *In the 12th International symposium on experimental robotics (ISER)*, **2010**.
- [24] Lowe, D. G., Distinctive image features from scale invariant keypoints, *International Journal of Computer Vision*, 60, 91–110, **2004**.
- [25] Fischler, M. A., Bolles, R. C., Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM*, 24, 381–395, **1981**.
- [26] Segal, A., Haehnel, D., Thrun, S., Generalized-ICP, *Robotics: Science and Systems*, 5, 168–176, **2009**.
- [27] Grisetti, G., Grzonka, S., Stachniss, C., Pfaff, P., Burgard, W., Efficient estimation of accurate maximum likelihood maps in 3D, *IEEE International Conference on Intelligent Robots and Systems*, 3472–3478, **2007**.
- [28] Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D., RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments, *International Journal of Robotics Research*, 31, 647–663, **2012**.
- [29] Rosten, E., Drummond, T., Machine learning for high-speed corner detection, *Lecture Notes in Computer Science*, 3951 LNCS, 430–443, **2006**.
- [30] Calonder, M., Lepetit, V., Fua, P., Keypoint signatures for fast learning and recognition, *Lecture Notes in Computer Science*, 5302 LNCS, 58–71, **2008**.
- [31] Nister, D., Stewenius, H., Scalable recognition with a vocabulary tree, *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, 2161–2168, 2, **2006**.
- [32] Konolige, K., Sparse Bundle Adjustment, *Bmvc*, 102.1-102.11, **2010**.



- [33] Engelhard, N., Endres, F., Hess, J., Sturm, J., Burgard, W., Real-time 3D visual SLAM with a hand-held RGB-D camera, *In Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum*, **2011**.
- [34] Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., Speeded-up robust features (SURF), *Computer vision and image understanding*, 110, 346–359, **2008**.
- [35] Grisetti, G., Kümmerle, R., Stachniss, C., Frese, U., Hertzberg, C., A Lazy, Online, and Hierarchical Optimization Approach for 2D and 3D Pose-Graphs Operating on Manifolds, *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 273–278, **2010**.
- [36] Fioraio, N., Konolige, K., Realtime Visual and Point Cloud SLAM, *Proc. of the RGB-D workshop on advanced reasoning with depth cameras at robotics: Science and Systems Conf.(RSS)*, Vol 27, **2011**.
- [37] Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W., G2o: A general framework for graph optimization, *Proceedings - IEEE International Conference on Robotics and Automation*, 3607–3613, **2011**.
- [38] Huang, A. S. *et al.*, Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera, *Proceedings of the International Symposium on Robotics Research (ISRR)*, 1–16, **2011**.
- [39] Mei, C., Sibley, G., Cummins, M., Newman, P. M., Reid, I. D., A Constant-Time Efficient Stereo SLAM System, *BMVC*, 1–11, **2009**.
- [40] Howard, A., Real-time stereo visual odometry for autonomous ground vehicles, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3946–3952, **2008**.
- [41] Horn, B. K. P., Closed-form solution of absolute orientation using unit quaternions, *JOSA A*, 4, 629–642, **1987**.
- [42] Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., Burgard, W., An evaluation of the RGB-D SLAM system, *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 1691–1696, **2012**.
- [43] Rublee, E., Rabaud, V., Konolige, K., Bradski, G., ORB: An efficient alternative to SIFT or SURF, *Proceedings of the IEEE International Conference on Computer Vision*, 2564–2571, **2011**.
- [44] Endres, F., Hess, J., Sturm, J., Cremers, D., Burgard, W., 3-D mapping with an RGB-D camera, *IEEE Transactions on Robotics*, 30, 177–187, **2014**.
- [45] Maier, R., Sturm, J., Cremers, D., Submap-based bundle adjustment for 3D reconstruction from RGB-D data, *Lecture Notes in Computer Science*, 8753, 54–65, **2014**.
- [46] Labbé, M., Michaud, F., Online global loop closure detection for large-scale multi-session graph-based SLAM, *IEEE International Conference on Intelligent Robots and Systems*, 2661–2666, **2014**.
- [47] Mur-Artal, R., Tardos, J. D., ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras, *IEEE Transactions on Robotics*, 33, 1255–1262, **2017**.
- [48] Gálvez-López, D., Tardós, J. D., Bags of binary words for fast place

- recognition in image sequences, *IEEE Transactions on Robotics*, 28, 1188–1197, **2012**.
- [49] Steinbrucker, F., Real-time visual odometry from dense RGB-D images, *Computer Vision Workshops (ICCV Workshops), IEEE International Conference on*, 719–722, **2011**.
- [50] Kerl, C., Cremers, D., Robust Odometry Estimation for RGB-D Cameras, *Robotics and Automation (ICRA), IEEE International Conference on*, 3748–3754, **2013**.
- [51] Kerl, C., Sturm, J., Cremers, D., Dense visual SLAM for RGB-D cameras, *IEEE International Conference on Intelligent Robots and Systems*, 2100–2106, **2013**.
- [52] Bylow, E., Sturm, J., Kerl, C., Kahl, F., Cremers, D., Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions, *Robotics: Science and Systems, Vol 2*, **2013**.
- [53] Sturm, J., Bylow, E., Kerl, C., Kahl, F., Cremers, D., Dense tracking and mapping with a quadcopter, *International Archives of Photogrammetry and Remote Sensing*, 4–6, **2013**.
- [54] Newcombe, R. A. *et al.*, KinectFusion: Real-Time Dense Surface Mapping and Tracking, *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, 127–136, **2011**.
- [55] Whelan, T., Kaess, M., Fallon, M., Kintinuous: Spatially extended kinectfusion, *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, 7, **2012**.
- [56] Whelan, T., Johannsson, H., Kaess, M., Leonard, J. J., McDonald, J., Robust real-time visual odometry for dense RGB-D mapping, *Robotics and Automation (ICRA), IEEE International Conference on*, 5724–5731, **2013**.
- [57] Whelan, T., Kaess, M., Johannsson, H., Fallon, M. F., Leonard, J. J., McDonald, J. B., Real-time Large Scale Dense RGB-D SLAM with Volumetric Fusion, *International Journal of Robotics Research*, **2015**.
- [58] Whelan, T., Kaess, M., Leonard, J. J., McDonald, J., Deformation-based loop closure for large scale dense RGB-D SLAM, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 548–555, **2013**.
- [59] Stückler, J., Behnke, S., Multi-resolution surfel maps for efficient dense 3D modeling and tracking, *Journal of Visual Communication and Image Representation*, 25, 137–147, **2014**.
- [60] Whelan, T., Salas-Moreno, R. F., Glocker, B., Davison, A. J., Leutenegger, S., ElasticFusion: Real-time dense SLAM and light source estimation, *International Journal of Robotics Research*, 35, 1697–1716, **2016**.
- [61] Glocker, B., Shotton, J., Criminisi, A., Izadi, S., Real-time RGB-D camera relocalization via randomized ferns for keyframe encoding, *IEEE transactions on visualization and computer graphics*, 21, 571–583, **2015**.
- [62] Du, H. *et al.*, Interactive 3D modeling of indoor environments with a consumer depth camera, *Proceedings of the 13th international conference on Ubiquitous computing*, 75–84, **2011**.

- [63] Meilland, M., Comport, A. I., Super-resolution 3D tracking and mapping, *Robotics and Automation (ICRA), IEEE International Conference on*, 5717–5723, **2013**.
- [64] Meilland, M., Comport, A. I., On unifying key-frame and voxel-based dense visual SLAM at large scales, *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, 3677–3683, **2013**.
- [65] Hu, G., Huang, S., Zhao, L., Alempijevic, A., Dissanayake, G., A robust rgb-d slam algorithm, *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, 1714–1719, **2012**.
- [66] Taylor, C. J., Cowley, A., Kettler, R., Ninomiya, K., Gupta, M., Niu, B., Mapping with depth panoramas, *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, 6265–6272, **2015**.
- [67] Handa, A., Whelan, T., McDonald, J., Davison, A. J., A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM, *Robotics and automation (ICRA), IEEE international conference on*, 1524–1531, **2014**.
- [68] Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D., A benchmark for the evaluation of RGB-D SLAM systems, *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, 573–580, **2012**.
- [69] Guclu, O., Can, A. B., A Comparison of Feature Detectors and Descriptors in RGB-D SLAM Methods, *Image Analysis and Recognition*, Springer, 297–305, **2015**.
- [70] Nistér, D., Naroditsky, O., Bergen, J., Visual odometry, *Computer Vision and Pattern Recognition (CVPR), Proceedings of the 2004 IEEE Computer Society Conference on*, 1, **2004**.
- [71] Leutenegger, S., Chli, M., Siegwart, R. Y., BRISK: Binary robust invariant scalable keypoints, *Computer Vision (ICCV), IEEE International Conference on*, 2548–2555, **2011**.
- [72] Shi, J., Tomasi, C., Good features to track, *Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society Conference on*, 593–600, **1994**.
- [73] Agrawal, M., Konolige, K., Blas, M. R., Censure: Center surround extremas for realtime feature detection and matching, *Computer Vision--ECCV 2008, Springer*, 102–115, **2008**.
- [74] Calonder, M., Lepetit, V., Strecha, C., Fua, P., Brief: Binary robust independent elementary features, *European conference on computer vision*, 778–792, **2010**.
- [75] Alahi, A., Ortiz, R., Vanderghenst, P., Freak: Fast retina keypoint, *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 510–517, **2012**.
- [76] Umeyama, S., Least-squares estimation of transformation parameters between two point patterns, *IEEE Transactions on pattern analysis and machine intelligence*, 13, 376–380, **1991**.
- [77] Open Source Computer Vision Library (OpenCV), <https://opencv.org/>

(Mart, **2018**).

- [78] Harris, C., Stephens, M., A combined corner and edge detector., *Alvey vision conference*, Vol 15, **1988**.
- [79] Guclu, O., Can, A. B., Histogram Based Visual Place Recognition for Improving SLAM Performance, *Autonomous Robot Systems and Competitions (ICARSC), IEEE International Conference on*, **2016**.
- [80] Sivic, J., Zisserman, A., Video Google: A text retrieval approach to object matching in videos, *Computer Vision, Proceedings. Ninth IEEE International Conference on*, 1470–1477, **2003**.
- [81] Cummins, M., Newman, P., Appearance-only SLAM at large scale with FAB-MAP 2.0, *The International Journal of Robotics Research*, 30, 1100–1123, **2011**.
- [82] Angeli, A., Filliat, D., Doncieux, S., Meyer, J.-A., Fast and incremental method for loop-closure detection using bags of visual words, *Robotics, IEEE Transactions on*, 24, 1027–1037, **2008**.
- [83] Nicosevici, T., Garcia, R., Automatic visual bag-of-words for online robot navigation and mapping, *IEEE Transactions on Robotics*, 28, 886–898, **2012**.
- [84] Swain, M. J., Ballard, D. H., Color indexing, *International journal of computer vision*, 7, 11–32, **1991**.
- [85] Arandjelović, R., Zisserman, A., Three things everyone should know to improve object retrieval, *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2911–2918, **2012**.
- [86] Huang, J., Kumar, S. R., Mitra, M., Zhu, W.-J., Zabih, R., Image indexing using color correlograms, *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 762–768, **1997**.
- [87] Guclu, O., Can, A. B., Fast and Effective Loop Closure Detection to Improve SLAM Performance, *Journal of Intelligent & Robotic Systems*, 1–23, **2017**.
- [88] Muja, M., Lowe, D. G., Fast matching of binary features, *Computer and Robot Vision (CRV), Ninth Conference on*, 404–410, **2012**.
- [89] Muja, M., Lowe, D. G., Scalable nearest neighbor algorithms for high dimensional data, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36, 2227–2240, **2014**.
- [90] Arun, K. S., Huang, T. S., Blostein, S. D., Least-squares fitting of two 3-D point sets, *IEEE Transactions on pattern analysis and machine intelligence*, 698–700, **1987**.

## ÖZGEÇMİŞ

### Kimlik Bilgileri

Adı Soyadı : Oğuzhan GÜÇLÜ  
Doğum Yeri : Bayburt  
Medeni Hali : Bekâr  
E-posta : oguzhanguclu86@gmail.com  
Adresi : Hacettepe Üniversitesi Bilgisayar Mühendisliği Bölümü  
Beytepe 06800 Ankara / Türkiye

### Eğitim

Lisans : Hacettepe Üniversitesi Mühendislik Fakültesi  
Bilgisayar Mühendisliği Bölümü, Ankara (2004 - 2009)  
Yüksek Lisans : -  
Doktora : Hacettepe Üniversitesi Fen Bilimleri Enstitüsü  
Bilgisayar Mühendisliği Anabilim Dalı, Ankara (2010 - 2018)

### Yabancı Dil ve Düzeyi

İngilizce : 82.5 (ÜDS 2009 - Bahar)

### İş Deneyimi

Araştırma Görevlisi : Hacettepe Üniversitesi Bilgisayar Mühendisliği (2011 - 2016)  
Yazılım Mühendisi : Sidus Innovations (2016 - Halen)

### Deneyim Alanları

Robotik, Bilgisayarlı Görü, Eş Zamanlı Konum Belirleme ve Haritalama, Makine Öğrenmesi

### Tezden Üretilmiş Projeler ve Bütçeleri

-

### Tezden Üretilmiş Yayınlar

- Oğuzhan Güçlü, Ahmet Burak Can, Fast and Effective Loop Closure Detection to Improve SLAM Performance, *Journal of Intelligent & Robotic Systems*, 1–23, **2017**.
- Oğuzhan Güçlü, Ahmet Burak Can, Histogram Based Visual Place Recognition for Improving SLAM Performance, *Autonomous Robot Systems and Competitions (ICARSC), IEEE International Conference on*, 174-180, **2016**.
- Oğuzhan Güçlü, Ahmet Burak Can, A Comparison of Feature Detectors and Descriptors in RGB-D SLAM Methods, *Image Analysis and Recognition*, Springer, 297–305, **2015**.

#### **Tezden Üretilmiş Tebliğ ve/veya Poster Sunumu ile Katıldığı Toplantılar**

- IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), **2016**.



HACETTEPE ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
YÜKSEK LİSANS/DOKTORA TEZ ÇALIŞMASI ORJİNALLİK RAPORU

HACETTEPE ÜNİVERSİTESİ  
FEN BİLİMLER ENSTİTÜSÜ  
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI BAŞKANLIĞI'NA

Tarih: 05/06/2018

Tez Başlığı / Konusu: **RGB-D Algılayıcılar Kullanılarak Eş Zamanlı Konum Belirleme ve Haritalama**

Yukarıda başlığı/konusu gösterilen tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler d) Sonuç kısımlarından oluşan toplam 139 sayfalık kısmına ilişkin, 05/06/2018 tarihinde ~~şahsım~~/tez danışmanım tarafından *Turnitin* adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 3'tür.

Uygulanan filtrelemeler:

- 1- Kaynakça hariç
- 2- Alıntılar hariç
- 3- 5 kelimedenden daha az örtüşme içeren metin kısımları hariç

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Uygulama Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini saygılarımla arz ederim.

05/06/2018

**Adı Soyadı:** Oğuzhan GÜÇLÜ  
**Öğrenci No:** N10164526  
**Anabilim Dalı:** Bilgisayar Mühendisliği  
**Programı:** Bilgisayar Mühendisliği  
**Statüsü:**  Y.Lisans  Doktora  Bütünleşik Dr.

**DANIŞMAN ONAYI**

UYGUNDUR.

  
Doç. Dr. Ahmet Burak CAN