

**ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**GÜNCEL GÖRÜNTÜYE DAYALI EŞ ZAMANLI KONUMLAMA VE
HARİTALAMA YÖNTEMLERİNİN FARKLI VERİ KÜMELERİ
ÜZERİNDEKİ PERFORMANS ANALİZİ**

Ömer Faruk YANIK

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

**ANKARA
2021**

Her hakkı saklıdır

ÖZET

Yüksek Lisans Tezi

GÜNCEL GÖRÜNTÜYE DAYALI EŞ ZAMANLI KONUMLAMA VE HARİTALAMA YÖNTEMLERİNİN FARKLI VERİ KÜMELERİ ÜZERİNDEKİ PERFORMANS ANALİZİ

Ömer Faruk YANIK

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Hakkı Alparslan ILGIN

Eş Zamanlı Konumlama ve Haritalama (EZKH), üzerinde çeşitli duyargalar bulunan bir robotun konumunu ve oryantasyonunu belirleme ile robotun bulunduğu çevrenin bir modelini (haritasını) eşzamanlı oluşturma problemi ile ilgilenmektedir. Robotun oryantasyonunu, hareketini ve bulunduğu çevreyi algılama görevini üstlenen duyargalar, EZKH çözümünün birincil unsurunu oluşturmaktadır. Düşük maliyetleri ve zengin veri içerikleri sebebiyle kameralar EZKH çözümlerinde gittikçe daha fazla tercih edilen duyargalar olmaktadır. Kameralar ile gerçekleştirilen EZKH çözümlerine Görsel-EZKH adı verilmektedir.

Kameralar ucuz duyargalar olmakla birlikte, gerçek zamanlı işlemler için yüksek işlem gücüne sahip donanımlara ihtiyaç duymaktadır. Bununla birlikte donanımların yüksek işlem gücü, mobil robotların güç kısıtları ile de sınırlandırılmaktadır. Bu sebeple literatürde bulunan Görsel-EZKH yöntemlerinin işlem yüküne ve güç tüketimine etkisinin karşılaştırmaya açık hususlar olduğu değerlendirilmektedir.

Bu çalışma kapsamında NVIDIA Jetson TX1 donanımı kullanılarak, ARM mimarisinde bir Merkezi İşlem Birimi (MİB) üzerinde çalıştırılmaya uygun olan ORB-SLAM2, Döngü Kapamalı Doğrudan Seyrek Odometri (DDSO) ve Doğrudan Seyrek Odometri (DSO) yöntemlerinin TUM Tek Kameralı Görsel Odometri, EUROC, ICL-NUIM ve KITTI veri kümeleri üzerindeki performanslarının karşılaştırılması amaçlanmıştır. Performans kriteri olarak hız, mutlak yörünge hatası, Rastgele Erişimli Bellek (REB) kullanımı, takas hafızası kullanımı, MİB kullanımı, Grafik İşlem Birimi (GİB) güç tüketimi ve toplam güç tüketimi parametreleri incelenmiştir.

Elde edilen sonuçlar incelendiğinde, ORB-SLAM2'nin, işlemci yükü ve yörünge takip hatası açısından en iyi yöntem olduğu, GİB güç tüketimi ve toplam güç tüketimi açısından bakıldığında DSO'nun beklendiğinden daha fazla maliyetli olduğu, REB kullanımı açısından ise en zorlayıcı yöntemin DDSO olduğu belirlenmiştir.

Eylül 2021, 75 sayfa

Anahtar Kelimeler: Robotik, Konumlama, Haritalama, Kapalı Ortam Navigasyonu, Görsel Navigasyon, EZKH, Hesaplama Maliyeti

ABSTRACT

Master Thesis

PERFORMANCE ANALYSIS OF STATE OF THE ART VISUAL SIMULTANEOUS LOCALIZATION AND MAPPING METHODS ON DIFFERENT DATASETS

Ömer Faruk YANIK

Ankara University
Graduate School of Natural and Applied Sciences
Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Hakkı Alparslan ILGIN

Simultaneous Localization and Mapping (SLAM) deals with the problem of determining the position and orientation of a mobile robot with various on-board sensors and simultaneously creating a map of the environment. Sensors perceiving the robot's orientation, motion and the environment of the robots are the primary elements of the SLAM. Because of their low prices and being rich data sources, cameras are being the more preferred sensors for Concurrent Mapping and Localization called as Visual-SLAM.

Although cameras are inexpensive, they require high processing performance for real-time operations. However, the high processing performance of the hardware is also limited by the power constraints of robots. For this reason, it is considered that the effect of Visual-SLAM methods on the processing load and power consumption of autonomous robots is open to comparison.

In this study, using NVIDIA Jetson TX1, it is aimed to compare the performances of ORB-SLAM2, Direct Sparse Odometry (DSO) and DSO with Loop Closure (LDSO) methods on TUM Monocular Visual Odometry, EUROC, ICL-NUIM and KITTI datasets. Speed, Absolute Trajectory Error (ATE), Random Access Memory (RAM), Swap Memory and Central Processing Unit (CPU) usage, Graphic Processing Unit (GPU) power consumption and total power consumption were examined as performance criteria.

Results showed that ORB-SLAM2 is the best method in terms of processor load and absolute trajectory error; DSO is costlier than expected in terms of GPU power consumption and total power consumption; and LDSO is the most challenging method in terms of RAM usage.

September 2021, 75 pages

Key Words: Robotics, Localization, Mapping, Indoor Navigation, Visual Navigation, SLAM, Computational Cost

ÖNSÖZ ve TEŞEKKÜR

Lisans döneminde olduğu gibi Yüksek Lisans tez çalışmalarımda da bilgi ve tecrübeleri ile desteklerini ve güvenini her zaman yanımda hissettiren, yönlendirmeleriyle çalışmamın şekillenmesine öncülük eden değerli danışman hocam Sn. Doç. Dr. Hakkı Alparslan ILGIN'a;

Çalışmalarım sırasında alan ile ilgili sorularımı karşılıksız bırakmayan Sn. Dr. Oğuzhan GÜÇLÜ'ye,

Yaşamım boyunca beni destekleyen, arkamda olduklarını her zaman hissettiğim, bu günlere gelmeme vesile olan değerli aileme;

Çalışmalarımda ve tez yazım sürecinde desteğini esirgemeyen, bu süreçte beni teşvik ve idare eden kıymetli eşime,

En derin duygularımla teşekkür ederim.

Ömer Faruk YANIK

Ankara, Eylül 2021

İÇİNDEKİLER

TEZ ONAY SAYFASI

ETİK.....	i
ÖZET.....	ii
ABSTRACT	iii
ÖNSÖZ ve TEŞEKKÜR.....	iv
KISALTMALAR DİZİNİ	vii
ŞEKİLLER DİZİNİ	ix
ÇİZELGELER DİZİNİ	xiv
1. GİRİŞ	1
2. EŞ ZAMANLI KONUMLAMA VE HARİTALAMA.....	4
2.1 EZKH Çözümünde Çevresel Etkenler	5
2.1.1 Duyarga doğruluğu	5
2.1.2 Ortamın büyüklüğü.....	7
2.1.3 Çevredeki dinamik unsurlar	7
2.2 EZKH'nin Olasılıksal Tanımı	8
2.3 Haritalama	10
2.4 Güncel Görsel EZKH Çalışmaları.....	11
3. MATERYAL VE YÖNTEM.....	19
3.1 Donanım Parametreleri	21
3.2 Karşılaştırılan EZKH Yöntemleri	23
3.2.1 ORB-SLAM2 yöntemi.....	23
3.2.2 Doğrudan Seyrek Odometri (DSO) yöntemi	24
3.2.3 Döngü Kapatmalı Doğrudan Seyrek Odometri (DDSO) yöntemi	25
3.3 Kullanılan Veri Kümeleri	25
3.3.1 ICL-NUIM veri kümesi	25

3.3.2	EUROC veri kümesi.....	28
3.3.3	KITTI veri kümesi	29
3.3.4	TUM TGO veri kümesi.....	30
3.4	Karşılaştırma Metriği	31
3.4.1	Mutlak Yörünge Hatası (MYH).....	31
3.4.2	Çerçeve başına çözümleme süresi	31
3.4.3	Rastgele Erişimli Bellek (REB) kullanımı.....	31
3.4.4	Merkezi İşlem Birimi (MİB) kullanımı	31
3.4.5	Grafik İşlem Birimi (GİB) güç tüketimi	31
3.4.6	Güç tüketimi	31
4.	EZKH YÖNTEMLERİNİN KARŞILAŞTIRILMASI.....	33
4.1	Mutlak Yörünge Hatası	33
4.2	Çerçeve Başına İşlem Süresi.....	40
4.3	Rastgele Erişimli Bellek Kullanımı.....	42
4.4	Merkezi İşlemci Birimi Kullanımı	50
4.5	Toplam Güç Tüketimi.....	54
4.6	Grafik İşlemci Birimi Güç Tüketimi	58
5.	TARTIŞMA VE SONUÇ.....	67
	KAYNAKLAR	70
	ÖZGEÇMİŞ.....	75

KISALTMALAR DİZİNİ

AÖB	Ataletsel Ölçüm Birimi
ATAGA	Ana Kare Tabanlı Görsel Ataletsel EZKH (ing. Open Keyframe Base Visual Inertial SLAM, OKVIS)
ÇDF	Çoklu Duyarga Füzyonu (İng. Multiple Sensor Fusion, MSF)
ÇDKKF	Çoklu Durum Kısıtlaması Kalman Filtresi (ing. Multi-State Constraint Kalman Filter, MSCKF)
ÇPTH	Çift Kameralı Paralel Takip ve Haritalama
DDSO	Döngü Kapamalı Doğrudan Seyrek Odometri (ing. Direct Sparse Odometry with Loop Closure, LDSO)
DSO	Doğrudan Seyrek Odometri (Direct Sparse Odometry)
DTKDK	Düzenlenmiş Tek Kameralı Derinlik Kestirimi (ing. Regularized Monocular Depth Estimation, REMODE)
ESA	Evrensel Seri Arayüz (ing. Universal Serial Bus, USB)
EUROC	European Robotics Challenge
EZKH	Eş Zamanlı Konumlama ve Haritalama (ing. Simultaneous Localization and Mapping,
GDTK	Geniş Ölçek Doğrudan Tek Kameralı (ing. Large-Scale Direct Monocular, LSD) SLAM)
GGAO	Gülbüz Görsel Ataletsel Odometri (ing. Robust Visual Inertial Odometry, ROVIO)
GİB	Grafik İşlem Birimi (ing. Graphics Processing Unit, GPU)
GTSAM	Georgia Tech Smoothing and Mapping
ICL-NUIM	Imperial College London and National University of Ireland Maynooth

ITEM	Işık ile Tespit ve Mesafelendirme (ing. Light Detection and Ranging, LIDAR)
KGO	Kısmi Doğrudan Görsel Odometri (ing. Semi-Direct Visual Odometry, SVO)
KITTI	Karlsruhe Institute of Technology and Toyota Technological Institute
KKS	Küresel Konumlama Sistemi (ing. Global Positioning System, GPS)
kok	Kök-Ortalama-Kare (Root-Mean-Square, rms)
KYM-D	Kırmızı-Yeşil-Mavi – Derinlik (Red-Green-Blue-Depth, RGB-D)
MB	Megabyte
MİB	Merkezi İşlem Birimi (ing. Central Processing Unit, CPU)
MPI	Max Planck Institute
MYH	Mutlak Yörünge Hatası (ing. Absolute Trajectory Error, ATE)
NYU-Depth	New York Üniversitesi – Derinlik (New York University - Depth)
PTH	Paralel Takip ve Haritalama (ing. Parallel Tracking and Mapping, PTAM)
RİS	Robotik İşletim Sistemi (ing. Robotic Operating System, ROS)
REB	Rastgele Erişimli Bellek (ing. Random Access Memory, RAM)
TGO	Tek Kameralı Görsel Odometri
TUM	Technical University of Munich
YÇÇA	Yüksek Çözünürlüklü Çoklu Ortam Arayüzü (ing. High Definition Multimedia Interface, HDMI)
YPPTH	Yoğun Parçalı Düzlemsel Takip ve Haritalama (ing. Dense Piecewise Planar Tracking and Mapping, DPPTAM)

ŞEKİLLER DİZİNİ

Şekil 2.1 EZKH Uygulama Alanlarına Örnekler	4
Şekil 2.2 KYM-D (Microsoft Kinect) ve Çift Gözlü Kamera (ZED) Örnekleri.....	6
Şekil 2.3 Dinamik Ortamlarda EZKH (macrovector.com, 2021)	8
Şekil 2.4 EZKH Problemi (Whyte ve Bailey 2006).....	9
Şekil 2.5. KITTI veri kümesini oluşturmak için kullanılan platform ve teçhizatları (Geiger vd. 2013)	13
Şekil 2.6 EUROC veri toplama platformu	13
Şekil 2.7 ICL-NUIM livingRoom0 video dizisine ait farklı açılardan ortam geometrisi (Handa vd. 2014).....	14
Şekil 2.8 TUM TGO video dizilerine ait video çerçeve örnekleri (Engel vd. 2016).....	14
Şekil 2.9 Sintel veri kümesine ait video çerçeve örnekleri (Buttler vd. 2012)	15
Şekil 2.10 Tsukuba veri kümesi (Silberman vd. 2012a)	15
Şekil 2.11 NYU-Depth veri kümesi ev ortamı.....	16
Şekil 2.12 Tek kameralı ORB-SLAM ve DPPTAM ile çift kameralı ZedFU ve RTAB-Map yöntemleri analiz robotu (İbragimov ve Afanasyev 2017).....	17
Şekil 2.13 Filipenko ve Afansayev EZKH robotu	18
Şekil 3.1 Çalışma Süreç Diyagramı	20
Şekil 3.2 NVIDIA Jetson TX1	21
Şekil 3.3 NVIDIA Jetson TX1 Geliştirme Kiti	21
Şekil 3.4 Tegrastats Metin Dosyası.....	22
Şekil 3.5 Tegrastats Güç Verisi Tanımı	22
Şekil 3.6 ORB-SLAM2 kurulum adımları	23
Şekil 3.7 DSO - Yörünge dosyasında sütunlar arası boşluklar	25
Şekil 3.8 (a) livingRoom0 video dizisine ait bir video çerçevesi , (b) officeRoom0 video dizisine ait bir video çerçevesi	26
Şekil 3.9 ICL-NUIM livingRoom0 ve officeRoom0 video dizilerinin kartezyen koordinat sisteminde gösterimi	26
Şekil 3.10 “times.txt” metin dosyasındaki satır-sütun yapısı - [isim zamanAdımı]	27
Şekil 3.11 EUROC veri kümesi MH01, V102 ve V203 video dizilerine ait video çerçeveleri	28

Şekil 3.12 EUROC veri kümesi MH01, V102 ve V203 isimli video dizilerinin kartezyen koordinatta gösterimi	28
Şekil 3.13 KITTI veri kümesi 00, 03, 07 isimli video dizilerine ait video çerçevesi örnekleri	29
Şekil 3.14 KITTI veri kümesi 00, 03 ve 07 isimli video dizilerinin kartezyen koordinatta gösterimi.....	29
Şekil 3.15 TUM TGO veri kümesi 19, 29 ve 30 isimli video dizilerine ait video çerçeveleri	30
Şekil 3.16 TUM TGO veri kümesi 19, 29, 30 isimli video dizilerinin Kartezyen koordinatta gösterimi	30
Şekil 4.1 ICL-NUIM veri kümesi üzerinde DSO, DDSO ve ORB-SLAM2 yöntemleri ile elde edilen MYH değerleri (kok)	34
Şekil 4.2 livingRoom0 - DSO, DDSO ve ORB-SLAM2 yöntemleri ile elde edilen yörüngelerin gerçek yer verileri üzerindeki iz düşümleri	34
Şekil 4.3 officeRoom0 - DSO, DDSO ve ORB-SLAM2 yöntemleri ile elde edilen yörüngelerin gerçek yer verileri üzerindeki iz düşümleri	35
Şekil 4.4 KITTI veri kümesine ait video dizileri üzerinde DSO, DDSO ve ORB-SLAM2 yöntemleri ile elde edilen MYH değerleri (kok).....	35
Şekil 4.5 KITTI veri kümesine ait 00, 03, 07 isimli video dizilerinde DSO, DDSO ve ORB-SLAM2 yöntemleri ile elde edilen yörüngelerin gerçek yer verisi üzerindeki iz düşümleri.....	36
Şekil 4.6 EUROC veri kümesine ait video dizileri üzerinde DSO, DDSO ve ORB-SLAM2 yöntemleri ile elde edilen MYH değerleri (kok).....	37
Şekil 4.7 EUROC veri kümesine ait MH01, V102, V203 isimli video dizilerinde DSO, DDSO ve ORB-SLAM2 yöntemleri ile elde edilen yörüngelerin gerçek yer verisi üzerindeki iz düşümleri.....	38
Şekil 4.8 TUM TGO veri kümesine ait video dizileri üzerinde DSO ve DDSO yöntemleri ile elde edilen MYH değerleri (kok).....	39
Şekil 4.9 DSO ve DDSO yöntemleri ile TUM TGO veri kümesine ait 19, 29 ve 30 numaralı video dizileri üzerinde elde edilen yörüngenin gerçek yer verisi üzerindeki iz düşümü.....	40

Şekil 4.10 ICL-NUIM veri kümesine ait officeRoom0 ve livingRoom0 video dizileri üzerinde EZKH yöntemlerinin çözümleme süreleri (ms)	41
Şekil 4.11 KITTI veri kümesine ait video dizileri üzerinde EZKH yöntemlerinin çözümleme süreleri (ms)	41
Şekil 4.12 EUROC veri kümesine ait video dizileri üzerinde EZKH yöntemlerinin çözümleme süreleri (ms)	42
Şekil 4.13 TUM TGO veri kümesine ait video dizileri üzerinde DSO ve DDSO yöntemlerinin çözümleme süreleri (ms)	42
Şekil 4.14 ICL-NUIM veri kümesine ait officeRoom0 ve livingRoom0 video dizileri üzerinde EZKH yöntemlerinin REB Kullanımlarına etkileri (MB - kok)	43
Şekil 4.15 ICL-NUIM veri kümesine ait livingRoom0 video dizisi - DSO, DDSO ve ORB-SLAM2 yöntemlerinin oluşturduğu REB yükleri	43
Şekil 4.16 ICL-NUIM veri kümesine ait officeRoom0 video dizisi - DSO, DDSO ve ORB-SLAM2 yöntemlerinin oluşturduğu REB yükleri	44
Şekil 4.17 DSO, DDSO ve ORB-SLAM2 yöntemlerinin KITTI veri kümesine ait 00, 03 ve 07 isimli video dizilerinde çözümleme yaparken donanımda oluşan REB yükleri	44
Şekil 4.18 DSO, DDSO ve ORB-SLAM2 yöntemlerinin KITTI 00 isimli video dizisi üzerinde çözümleme yaparken donanımda oluşan REB yükü	45
Şekil 4.19 DDSO Düşük Hafıza Uyarısı - KITTI 00	46
Şekil 4.20 KITTI 00 video dizisi üzerinde DSO, DDSO ve ORB-SLAM2 yönteminin oluşturduğu takas (swap) hafıza yükleri	46
Şekil 4.21 EZKH yöntemlerinin EUROC veri kümesi üzerinde çözümleme yaparken donanım üzerinde oluşan REB yükü	47
Şekil 4.22 EUROC veri kümesine ait MH01 video dizisi için çözümleme süresince REB kullanım davranışları	47
Şekil 4.23 TUM TGO veri kümesine ait video dizileri üzerinde DSO ve DDSO yöntemlerinin donanımın REB kullanımına etkileri	48
Şekil 4.24 TUM TGO 19 isimli video dizisi üzerinde zamana bağlı REB kullanımı	48
Şekil 4.25 TUM TGO 29 isimli video dizisi üzerinde zamana bağlı REB kullanımı	49
Şekil 4.26 TUM TGO 30 isimli video dizisi üzerinde zamana bağlı REB kullanımı	49

Şekil 4.27 ICL-NUIM veri kümesi üzerinde EZKH yöntemlerinin çözümlemeleri esnasında donanımda oluşan MİB yükü (kok).....	50
Şekil 4.28 ICL-NUIM officeRoom0 video dizisi üzerinde EZKH yöntemlerinin çözümlemeleri esnasında donanımda oluşan zamana bağlı MİB yükleri	51
Şekil 4.29 ICL-NUIM livingRoom0 video dizisi üzerinde EZKH yöntemlerinin çözümlemeleri esnasında donanımda oluşan zamana bağlı MİB yükleri	51
Şekil 4.30 KITTI veri kümesine ait 00, 03 ve 07 isimli video dizileri üzerinde EZKH yöntemlerinin çözümlemesi esnasında donanımda oluşan MİB yükleri (kok).....	52
Şekil 4.31 EUROC veri kümesine ait video dizileri üzerinde EZKH yöntemlerinin çözümlemesi esnasında donanımda oluşan MİB yükleri (kok)	53
Şekil 4.32 TUM TGO veri kümesine ait video dizileri üzerinde EZKH yöntemlerinin getirdiği MİB yükleri (kok).....	53
Şekil 4.33 ICL-NUIM veri kümesi üzerinde DSO, DDSO ve ORB-SLAM2 yöntemlerinin güç tüketimine etkileri (kok)	54
Şekil 4.34 ICL-NUIM officeRoom0 video dizisi üzerinde DSO, DDSO ve ORB-SLAM2 yöntemlerinin zamana bağlı güç tüketimine etkisi	55
Şekil 4.35 ICL-NUIM livingRoom0 video dizisi üzerinde DSO, DDSO ve ORB-SLAM2 yöntemlerinin zamana bağlı güç tüketimine etkisi	55
Şekil 4.36 KITTI veri kümesine ait video dizileri üzerinde EZKH yöntemlerinin güç tüketimine etkisi (kok)	56
Şekil 4.37 KITTI 00 video dizisinin çözümlenmesi esnasında donanımın zamana bağlı güç tüketimi.....	56
Şekil 4.38 EUROC veri kümesine ait video dizileri üzerinde DSO, DDSO ve ORB-SLAM2 yöntemlerinin donanımın güç tüketimine etkileri (mW - kok).....	57
Şekil 4.39 TUM TGO veri kümesine ait video dizileri üzerinde DSO, DDSO ve ORB-SLAM2 yöntemlerinin donanımın güç tüketimine etkileri (mW - rms).....	58
Şekil 4.40 ICL-NUIM veri kümesine ait officeRoom0 video dizisi üzerinde zamana GİB güç tüketim karşılaştırmaları	59
Şekil 4.41 ICL-NUIM veri kümesine ait livingRoom0 video dizisi üzerinde zamana GİB güç tüketim karşılaştırmaları	59
Şekil 4.42 KITTI veri kümesine ait video dizileri üzerinde çözümleme yapan EZKH yöntemlerinin GİB Güç Tüketimine etkileri (kok)	60

Şekil 4.43 KITTI veri kümesine ait 00 isimli video dizileri üzerinde EZKH yöntemlerinin GİB güç tüketimine zamana bağlı etkisi tüketimi	61
Şekil 4.44 KITTI veri kümesine ait 03 isimli video dizileri üzerinde EZKH yöntemlerinin GİB güç tüketimine zamana bağlı etkisi tüketimi	61
Şekil 4.45 KITTI veri kümesine ait 07 isimli video dizileri üzerinde EZKH yöntemlerinin GİB güç tüketimine zamana bağlı etkisi.....	62
Şekil 4.46 EUROC veri kümesine ait video dizileri üzerinde EZKH yöntemlerinin donanımın GİB güç tüketimine etkisi (mW - kok)	62
Şekil 4.47 EUROC veri kümesine ait MH01 isimli video dizisinin çözümlenmesi esnasında donanımın GİB Güç Tüketimi	63
Şekil 4.48 EUROC veri kümesine ait V102 isimli video dizisinin çözümlenmesi esnasında donanımın GİB Güç Tüketimi (mW)	63
Şekil 4.49 EUROC veri kümesine ait V203 isimli video dizisinin çözümlenmesi esnasında donanımın GİB Güç Tüketimi (mW)	64
Şekil 4.50 TUM TGO veri kümesi üzerinde DSO ve DDSO yöntemlerinin çözümlenmesi esnasında donanımın GİB güç tüketimi (mW - kok)	64
Şekil 4.51 TUM TGO veri kümesi 19 isimli video dizisi üzerinde DSO ve DDSO yöntemlerinin çözümlenmesi esnasında donanımın zamana bağlı GİB güç tüketimi	65
Şekil 4.52 TUM TGO veri kümesi 29 isimli video dizisi üzerinde DSO ve DDSO yöntemlerinin çözümlenmesi esnasında donanımın zamana bağlı GİB güç tüketimi (mW)	65
Şekil 4.53 TUM TGO veri kümesi 30 isimli video dizisi üzerinde DSO ve DDSO yöntemlerinin çözümlenmesi esnasında donanımın zamana bağlı GİB güç tüketimi (mW)	66

ÇİZELGELER DİZİNİ

Çizelge 2.1 Görsel EZKH Algoritmaları (Liang vd. 2016)	12
Çizelge 3.1 NVIDIA Jetson TX1 Donanım Özellikleri	21



1. GİRİŞ

Otonom hareket yeteneđi, sivil sistemlerden askeri sistemlere kadar birçok alanda gün getike temel gereksinimler arasına girmektedir. Otonom robotlar, günlük hayatta sürücüsüz otomobillerden temizlik robotlarına, silahlı insansız hava araçlarından, tüketici elektroniđine kadar birçok ürünle birlikte daha görünür ve konuşulur olmaya başlamıştır. Bu sistemlerin en belirgin ortak özelliđi, tanımlanan görevleri kısıtlı kullanıcı müdahalesi ile ya da kullanıcı müdahalesi olmaksızın gerçekleştirebilmeleridir.

Önceden bilinen ya da Küresel Konumlama Sistemi (KKS, Global Positioning System - GPS) gibi uydu sinyallerinin alınabildiđi ortamlarda, yol noktası tanımlamak gibi görece basit otonom hareket yeteneđi geleneksel uygulamalarla sağlanabilmektedir. KKS, Ataletsel Ölüm Birimi (AÖB, Inertial Measurement Unit – IMU), Işık ile Tespit ve Mesafelendirme (ITEM, Light Detecting and Ranging - LIDAR) vb. duyarılardan yararlanılarak konum bilgisi elde edilebilmektedir. Bu bilgiler yol noktası tanımlama, şerit takibi, geilen mesafenin tahmini, engellerden kaçınma ve mesafe ayarlama gibi işlevler için belirli bir standartta kullanılabilmektedir. Bu uygulamalarda, sahip olduđu algılayıcılar vasıtası ile konumsal ilişkisi bilinen robot, ilgili referans bilgileri kullanarak kendi konumunu kestirmektedir. Bununla birlikte bu yöntemler, yüksek hata payı sebebiyle robot tarafından KKS uydu sinyallerinin algılanamadıđı veya öncesinde herhangi bir bilgi sahibi olunmayan ortamlarda engellerden kaçma, anlık konumunu hesaplama ve çevrenin algılanması gibi işlevler açısından yeterli olmamaktadır. Önceden bilinmeyen ya da KKS sinyallerinin algılanamadıđı ortamlarda, robotun çevreye göre konumunu hesaplayabilmesi için farklı çözümlere ihtiyaç duyulmaktadır.

Otonom hareket yeteneđinin artırılması için hareket kestirimi, hedef takibi, KKS tabanlı yol noktası izleme gibi çeşitli yöntemler ve bu yöntemlere dayalı farklı yorumlar içeren birçok alışma bulunmaktadır. Ancak, önceden bilgi sahibi olunmayan ortamlarda bir robotun konumunun tayin edilebilmesi için öncelikle çevreye dair bir bilgi edinilmesi, çevreye dair bilgi edinirken de robotun konumunun hesaplanması gerekmektedir.

Konum belirleme yöntemlerinden biri olan odometri, sensörler kullanılarak robotun bir önceki konumuna göre yer deđiştirmesinin hesaplanması olarak tanımlanmaktadır. Tekerleđin boşa dönmesi ya da kayması gibi hata kaynaklarına sahip olan tekerlek

odometrisi, referans bir bilgiye sahip olunmayan ortamlarda, konumlandırmanın en basit yollarından biri olarak ortaya çıkmaktadır (Yousif vd. 2015). Tekerleğin kayması ya da boşa dönmesi gibi çeşitli hatalar, bir önceki konum bilgisindeki hata bir sonraki konum bilgisini de etkileyeceği için tekerlek odometrisinde toplam hatayı biriktirerek artırmaktadır. Ayrıca tekerlek odometrisi yalnızca tekerlekli araçlar için kullanılabilir. Bu kısıtların üstesinden gelebilmek için Lazer Odometri, Görsel Odometri ve Eş Zamanlı Konumlama ve Haritalama (EZKH, Simultaneous Localization and Mapping – SLAM) gibi farklı konumlandırma yöntemleri bulunmaktadır. Görsel odometri yönteminde, bir önceki konuma göre kameranın mevcut konumu hesaplanarak güzergâhın takip edilmesi sağlanmaktadır. EZKH’de ise geçilen tüm konum ve harita bir bütün olarak ele alınmaktadır (Yousif vd. 2015).

Son yıllarda önemi gittikçe artan EZKH, üzerinde duyargalar bulunan bir robotun eş zamanlı olarak bir yandan ortamdaki konumunu belirlerken diğer yandan çevresini gözlemleyerek ortamın bir haritasını oluşturması ile ilgilenmektedir. Yumurta-tavuk ilişkisi sebebiyle EZKH zor bir konu olarak ortaya çıkmaktadır (Huletski vd. 2015).

Robotun gerçekleştirdiği yer değiştirmenin hesaplanması (odometri) ve tekrar geçilen bir yerin anlaşılması (döngü kapama, loop closure) EZKH’nin temel bileşenlerini oluşturmaktadır (Güçlü, 2018). EZKH’nin, teorik olarak birçok şekilde formüle edilmesinin yanında, kapalı ortam robotlarından sualtı ve hava sistemlerine kadar birçok alanda uygulanma imkânı bulunmaktadır. Robotların gerçek anlamda otonom olabilmesini sağlayan EZKH probleminin çözümü, robotik alanının en önemli başarılarından biri olarak görülmektedir. (Whyte ve Bailey, 2006).

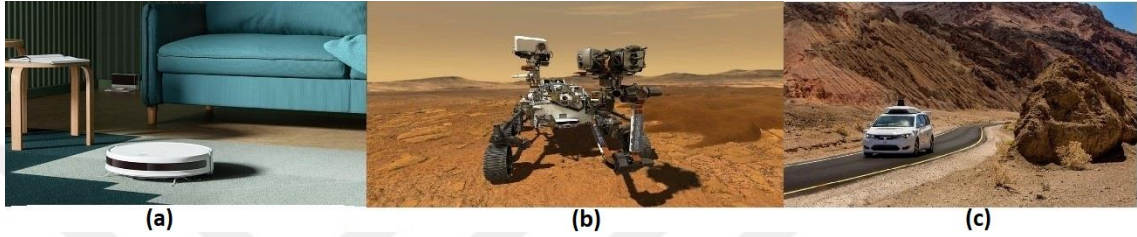
EZKH probleminin çözümü için geliştirilen yöntemlerin çeşitliliğinin yanında, bu yöntemlerin performans analizleri için kullanılan veri kümeleri de çeşitlilik göstermektedir. Bu veri kümeleri genellikle hareket yakalama gibi yöntemler ile laboratuvarlarda oluşturulmakta olup, iç ortam (indoor), dış ortam (outdoor), hareketli nesneler içeren dinamik ortamlar ya da görece durağan ortamlar gibi özelliklerle birbirlerinden ayrılmaktadır.

Tez çalışması kapsamında, literatürde yer alan ORB-SLAM2 (Mur-Artal ve Tard’os, 2017) , DDSO (Döngü Kapamalı Doğrudan Seyrek Odometri, Direct Sparse Odometry with Loop Closure, LDSO) (Gao vd. 2018) ve DSO (Doğrudan Seyrek Odometri, Direct

Sparse Odometry) (Engel vd. 2017) yöntemlerinin iç ortam ve dış ortam gibi bilgileri içermesi bakımından birbirinden ayrılan ve farklı laboratuvar ortamlarında oluşturulan ICL-NUIM (Handa vd. 2014), KITTI (Geiger vd. 2013), EUROOC (Burri vd. 2016) ve TUM TGO (Engel vd. 2016) veri kümeleri üzerindeki performansları NVIDIA Jetson TX1 donanımı kullanılarak karşılaştırılmıştır. Performans kriteri olarak doğruluk ve hız parametrelerinin yanında NVIDIA Jetson TX1 donanımının Merkezi İşlem Birimi (MİB, Central Processing Unit - CPU), Grafik İşlemci Birimi (GİB, Graphics Processing Unit- GPU), ve Rastgele Erişimli Bellek (REB, Random Access Memory - RAM) kullanımları gibi parametreler gözlenmiştir. Bu bilgilere ek olarak EZKH algoritmalarının güç tüketimlerine yönelik karşılaştırma yapılmıştır. Bölüm 2’de EZKH’ye ve veri kümelerine dair literatür bilgisi ile benzer çalışmalara, Bölüm 3’te ise tez çalışmasında kullanılan materyal ve deney yöntemine, Bölüm 4’te elde edilen bulgulara yer verilmiştir. Bölüm 5’te ise bulgulardan çıkarılan sonuçlar ortaya konulmuştur.

2. EŞ ZAMANLI KONUMLAMA VE HARİTALAMA

EZKH, üzerinde duyargalar bulunan bir robotun eş zamanlı olarak bir yandan ortamdaki konumunu belirlerken diğer yandan bulunduğu ortamın bir haritasını oluşturması problemi ile ilgilenmektedir. Sürücüsüz otomobillerden (Şekil 2.1c) (Xiaomi, xiaomi.com,2021) , uzay araçlarına(Şekil 2.1b) (NASA, nasa.gov, 2021) ve temizlik robotlarına ((Şekil 2.1a) (Xiaomi, xiaomi.com, 2021) kadar birçok alanda çözülmesi gereken temel sorunlardan birisi olarak ortaya çıkmaktadır.



Şekil 2.1 EZKH Uygulama Alanlarına Örnekler

Robotun çevreye dair kusursuz bir ön bilgiye sahip olması durumunda robotun konumu nispeten kolay bir şekilde hesaplanabilecektir. Ya da tam tersine, robot konumunun kusursuz bir şekilde bilinmesi durumunda, bulunduğu çevrenin gürbüz bir modeli oluşturulabilecektir. Ancak EZKH problemi kapsamında ise üzerinde duyargalar bulunan robot, bulunduğu çevreye ilişkin ön bilgiye sahip değildir. Robotun konumu ile çevrenin haritasının eş zamanlı oluşturulması ve kusursuz bir bilgi elde edilememesi sebebiyle EZKH probleminin çözümünde olasılıksal yöntemler kullanılmaktadır (Güçlü, 2018).

EZKH probleminin çözümünde duyarga ölçümleri, harita gösterimi, robot dinamikleri, çevresel etkenler ve robota ait kontrol sistemlerinin sürekli hesaba katılıyor olması gerekmektedir. Bu çok yönlü işlemlerin gerçek zamanlı olarak gerçekleştirilebilmesi için çoğunlukla Genişletilmiş Kalman Filtresi, Parçacık Filtresi ve Çizge Tabanlı yöntemler kullanılmaktadır (Chong vd. 2015).

Müteakip başlıklarda EZKH çözümünü etkileyen çevresel faktörlere, EZKH çözümünün matematiksel modeline, haritalama tekniklerine ve EZKH alanında gerçekleştirilen çalışmalara ilişkin bilgiler verilmektedir.

2.1 EZKH Çözümünde Çevresel Etkenler

Günlük hayatta otonom olması beklenen sistemlerin, olağan çevresel etkenler ile de başa çıkabilmesi beklenmektedir. Bununla birlikte EZKH probleminin çözümü kapsamında birçok zorlayıcı etken bulunmaktadır. Kullanılan duyargaların gürültüleri, robotun bulunduğu alanın büyüklüğü ve çevredeki dinamik unsurlar gibi etkiler EZKH probleminin çözümünde matematiksel ve donanımsal zorluklar ortaya çıkarmaktadır.

Çevrenin ve robotun durumuna ilişkin verinin güvenilirliğinin bir göstergesi olarak duyargaların doğruluğu EZKH probleminin çözümünde önemli bir faktör olarak ortaya çıkmaktadır. Ayrıca, bulunulan ortamın büyüklüğü duyarga kaynaklı hata birikimini artıracak için ortam büyüklüğü bir diğer problem olarak kendini göstermektedir. Çevrenin modelini sürekli değiştiren hareketli nesneler ise EZKH probleminin çözümü için hesaba katılması gereken bir diğer çevresel etkidir. Bu etkenlere ilişkin detaylar alt başlıklarda verilmektedir.

2.1.1 Duyarga doğruluğu

Otonom hareket yeteneği için robotun kendi oryantasyonuna ve çevresine ilişkin bilgiyi olabildiğince kusursuz şekilde elde edebilmesi gerekmektedir (İlci ve Toth 2019). İnsansız otomobil ve uzay görevleri gibi otonom sistemlerde çevrenin ve robotun güvenliği için çevreye ilişkin bilginin derinliği de önem kazanmaktadır.

Robotun kendi oryantasyonunu, hareketini ve bulunduğu çevreyi algılama görevini üstlenen duyargalar, EZKH çözümünün birincil unsurunu oluşturmaktadır.

EZKH'nin matematiksel çözüm yöntemlerinin yanında, EZKH çözümünde kullanılan duyargalar da çeşitlilik göstermektedir. Mesafe ölçümü ile çevrenin taranması için kullanılan sonar ve LIDAR duyargalarından, tek gözlü, iki gözlü ve KYM-D (Kırmızı Yeşil Mavi- Derinlik, RGB-D-Red Green Blue –Depth) kameralara kadar farklı duyargalar EZKH çözümlerinde kullanılmaktadır. EZKH probleminin çözümünde tek bir duyargadan bilgi alınabileceği gibi, aynı türden çok sayıda duyarga veya birden farklı duyargaların kombinasyonları ile de çevreye ilişkin veriler elde edilebilmektedir.

Matematiksel modele girdi yapacak olan duyarga verilerinin gürbüzlüğü, oluşacak olan hataların toplanarak birikmesini de engelleyecektir. Ancak gerçek hayatta kusursuz bir

duyarga verisi elde etmek mümkün değildir. Bu sebeple EZKH probleminin çözümü duyarga hata ve gürültülerinin de üstesinden gelebilmelidir.

Literatürdeki EZKH çözümlerinde kullanılan en yaygın algılama yöntemleri, menzil tarama duyargalarına ve öznitelik (feature - landmark) tabanlı duyargalara dayanmaktadır (Chong vd. 2015). İvmeölçer ve mesafe ölçer ya da tekerlek çözücüsü (encoder) gibi mesafe ölçümüne dayanan duyargalar yalnızca alınan yol veya engele yaklaşıma dayalı veriler ile çevrenin tanınmasını sağlamakta ve konumda yüksek oranlı kaymalara sebep olmaktadır. Öznitelik tabanlı duyargalarda ise konumda kayma daha iyi ele alınabilmektedir.

Öznitelik tabanlı duyargalar arasında LIDAR ve kameralar sayılabilmektedir. Düşük gürültü seviyesine sahip olan LIDAR gibi duyargaların maliyetleri kameralara göre oldukça yüksektir (Yousif vd. 2015). Kameralar çevreye dair çok daha fazla bilgiyi sağlayabilmektedir. Ancak kameraya ait video çerçevelerini gerçek zamanlı çözümlemek için yüksek işlem gücüne sahip donanımlara ihtiyaç duyulmaktadır.

Çevre algılama birimi olarak kameraların kullanıldığı çözümler Görsel EZKH (Visual SLAM) olarak adlandırılmaktadır. Görsel-EZKH, çevrenin algılanması, robot konumunun tayin edilebilmesi ve maruz kalınan ortamın bir modelinin oluşturulabilmesi için kaynak olarak yalnızca kameralardan alınan görüntülerin kullanıldığı yöntemdir (Fuentes-Pacheco vd. 2015). Tek kameralı (Monocular) çözümlerin yanında, çift kameralı (Stereo) ve üzerinde kızılötesi derinlik algılayıcı bulunan kameraların kullanıldığı KYM-D EZKH çözümleri de dikkat çekmektedir. Şekil 2.2’de çift kamera ve KYM-D kameralara örnekler verilmektedir.



Şekil 2.2 KYM-D (Microsoft Kinect) ve Çift Gözlü Kamera (ZED) Örnekleri

2.1.2 Ortamın büyüklüğü

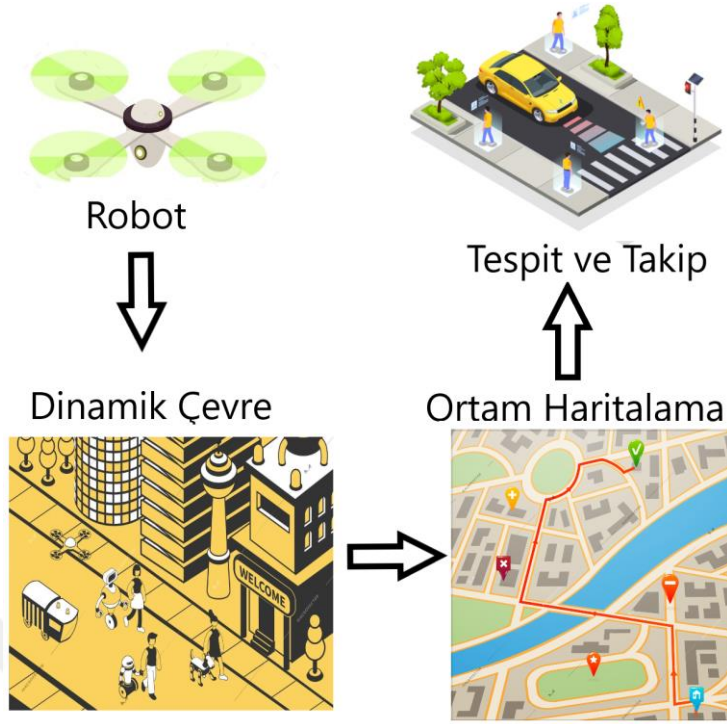
Duyarga doğruluğu ve donanım kabiliyetleri ile de ilişkili olarak EZKH probleminin çözümünde ele alınması gereken faktörlerden birisi de robotun bulunduğu ortamın büyüklüğüdür.

EZKH probleminin çözümü, dış ortamlara kıyasla referans bilgiye ulaşmanın zor olduğu nispeten küçük alanlara sahip iç ortamlarda daha ön plandadır. Ancak çevrenin anlamlandırılabilmesi ve böylece otonom hareket yeteneğinin artırılması için geniş alanlardan oluşan dış ortamlarda da EZKH probleminin çözülmesi, otonom sistemlerin temel gereksinimleri arasındadır (Cadena vd. 2016).

Robotun bulunduğu ortam büyüdükçe, duyarga doğruluğu ile ilişkili olarak hatalar birikimli olarak artmaktadır. Kamera gibi doğruluğu artırmak açısından zengin veri içeren duyargalarda ise görüş mesafesinin kısa olması ve işlem yükünün artırması sebebiyle zorluklar ortaya çıkmaktadır (Liang vd. 2016). Döngü Kapama işlemi için tutulan verilerin ilişkilendirilmesinin zorlaştığı, güç tüketiminin, REB kullanımının ve işlem yükünün arttığı geniş alanlar EZKH probleminin çözümü açısından üstesinden gelinmesi gereken konulardan biridir.

2.1.3 Çevredeki dinamik unsurlar

Gerçek bir otonom hareket kabiliyeti için robotun Şekil 2.3 ile verilen senaryoda olduğu gibi, hareketli ve durağan nesneler içeren ortamlarda kabiliyetlerini yerine getirebiliyor olması beklenmektedir. Ancak literatürdeki EZKH çözümüne yönelik çalışmalarda robotun bulunduğu ortamın hareketli nesneler içermeyen, görece durağan bir ortam olduğu kabul edilmektedir (Wang vd. 2007). EZKH problemi teorik olarak 1980'lerde ortaya atılmış olmakla birlikte (Whyte ve Bailey 2006), EZKH kapsamında ortamda bulunan hareketli nesnelere yönelik ilk öneriler Wang vd (2004), Wangsiripitak ve Murray (2009), Migliore vd. (2009) ile Lin ve Wang (2010) çalışmalarında ortaya atılmıştır (Pacheco vd. 2012).

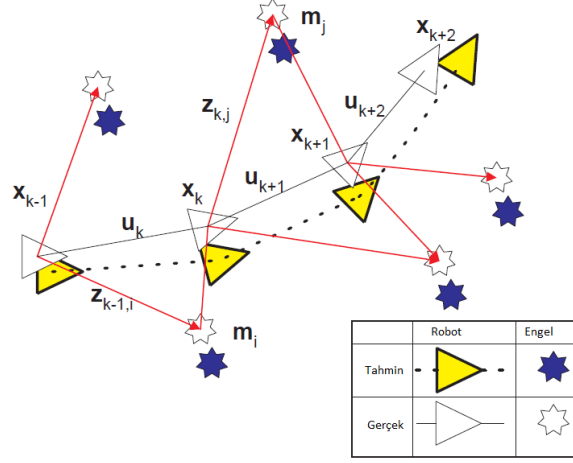


Şekil 2.3 Dinamik Ortamlarda EZKH (macrovector.com, 2021)

2.2 EZKH'nin Olasılıksal Tanımı

EZKH problemi kapsamında üzerinde duyargalar bulunan robot, bulunduğu çevreye ilişkin ön bilgiye sahip değildir. Robotun konumu ile çevrenin modelinin eş zamanlı oluşturulması, konum ve çevrenin modeline ilişkin kusursuz bilgiye sahip olunmaması sebebiyle EZKH probleminin çözümünde olasılıksal yöntemler kullanılmaktadır (Güçlü, 2018).

Şekil 2.4 ile EZKH problemini tanımlayan bir senaryo verilmektedir.



Şekil 2.4 EZKH Problemi (Whyte ve Bailey 2006)

x_k , robotun konumunu ve oryantasyonunu; u_k , k-1 anında, robotu x_k durumuna getirmek için uygulanan kontrol girdisini; m_i ortam haritasını oluşturan her bir engelin konumunu; z_{ik} , k anında robot tarafından yapılan gözlemler ifade etmektedir.

Ayrıca $\mathbf{X}_{0:k} = \{x_0, x_1, \dots, x_k\}$, robot geçmişte uğradığı konumlar; $\mathbf{U}_{0:k} = \{u_1, u_2, \dots, u_k\}$, kontrol girdilerinin geçmişi; m robotun bulunduğu ortamdaki tüm yer göstericiler; $\mathbf{Z}_{0:k} = \{z, z_1, \dots, z_k\}$ robot tarafından engeller üzerinde yapılan gözlemlerin tümü olmak üzere, EZKH probleminin çözümü, Eşitlik (1) ile verilen ve tüm süre boyunca hesaplanması gereken olasılıksal (P) yaklaşıma dayanmaktadır.

$$P(x_k, m \mid \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, x_0) \quad (1)$$

Eşitlik (1) ile verildiği üzere, robotun başlangıç konumu, kontrol vektörü ve haritanın unsurları olan engellere ait robot tarafından yapılan gözlemler bilindiğinde, robotun mevcut konumunun ve ortamın haritasının güncellenmesine ilişkin olasılık elde edilmektedir.

Bu olasılıksal tanımla birlikte, EZKH probleminin çözümünde birçok yöntem ortaya çıkmıştır. Başta gelen yöntemler Genişletilmiş Kalman Filtresi (GKF) ve Parçacık Filtresi yöntemleridir.

EZKH probleminin çözümünde kullanılan ve bir kilometre taşı olarak kabul edilen GKF yöntemi, verilerin zamanla birikmesinden kaynaklı olarak yüksek işlem maliyeti getirmektedir. Bu da GKF büyük ortamlarda gerçek zamanlı haritalama ve konumlama için kullanılmasının önüne geçmektedir (Korkmaz vd. 2018).

EZKH çözümünde kullanılan bir diğer yöntem ise HızlıEZKH (FastSLAM) (Thrun vd. 2004) algoritması ile ilk defa kullanılan Parçacık Filtresi yöntemidir. Detaylı bir şekilde Whyte ve Bailey'nin (2006) açıkladığı Parçacık Filtresi yöntemi, küçük boyutlu uzaylarda düzgün sonuçlar vermektedir. Parçacık Filtresi yöntemi, Rao-Blackwellisation uygulanarak büyük boyutlu uzaylarda da kullanılabilir.

Kalman ve parçacık filtresi yöntemleri robotun mevcut durumu ve çevrenin mevcut haritası ile ilgilenmektedir. Bu sebeple bu yöntemler Çevrimiçi EZKH olarak adlandırılmaktadır. Modern EZKH algoritmalarında çoğunlukla kullanılan yöntemlerinden birisi de Çizge Tabanlı yöntemdir. Robotun başlangıçtan itibaren gerçekleştirdiği hareketin ve haritanın tümü ile ilgilenilen Çizge Tabanlı yöntemler ise Tam-EZKH (Full-SLAM) olarak adlandırılmaktadır (Grisetti vd. 2010).

2.3 Haritalama

EZKH çözümünün bir diğer adımı ise ortamın haritasının çıkarılmasıdır. EZKH çözümlerinde kullanılan haritalar çevreyi tanımlamak üzere kullanılacak olan herhangi bir model olabilir. Bu modeller topolojik ya da geometrik haritalar şeklinde oluşturulabilir. Topolojik haritalar, mesafe bilgisi olmaksızın çevredeki ilişkileri göstermektedirler. Geometrik haritalar ise, bölgeler arasındaki bağlantıların yanı sıra mesafe bilgilerini de içerirler. Diğer yandan, haritalar nitelik tabanlı ya da hacimsel olabileceği gibi 2 boyutlu ya da 3 boyutlu olarak oluşturulabilirler (Güçlü, 2018).

2.4 Güncel Görsel EZKH Çalışmaları

EZKH çözümünde kullanılabilecek duyargalar çeşitlilik göstermektedir. Bunun yanında düşük maliyetleri, tek bilgi kaynağı olması ve zengin bilgi içeriği gibi sebeplerle kameralar EZKH probleminin çözümünde çokça tercih edilen duyargalar arasındadır. Tek kameralı, çift kameralı ve kızılötesi derinlik algılayıcılı kamera çözümleri literatürde mevcuttur. Kameraların kullanıldığı EZKH yöntemlerine Görsel EZKH (Visual SLAM - VSLAM) adı verilmektedir.

Kameraların kullanıldığı yöntemler genel olarak tüm görüntünün kullanıldığı doğrudan (direct) ve özniteliklere odaklanan öznitelik tabanlı (feature-based) yöntemler olmak üzere ikiye ayrılmaktadır (Filipenko ve Afanasyev 2018). Ayrıca anahtar çerçeve seçimi, paralel takip ve haritalama özellikleri açısından öznitelik tabanlı yöntemlere benzeyen ancak doğrudan yöntemlerdeki gibi piksel yoğunluklarını takip eden yarı doğrudan (Semi-direct) yöntemler de bulunmaktadır. Oluşturulan harita detaylarının yoğunluğuna göre ise seyrek (sparse) ya da yoğun (dense) olarak sınıflandırılabilir. Döngü kapama (geçilen yeri tanıma) özelliği içeren görsel EZKH yöntemleri olmakla birlikte, Odometri olarak adlandırılan ve döngü kapama özelliği bulunmayan algoritmalar da bulunmaktadır.

Bugüne kadar tek kameralı ve çok kameralı duyarga konfigürasyonlarında birçok çalışma gerçekleştirilmiştir. Bunlar arasında Çizelge 2.1 ile verildiği üzere öznitelik temelli Paralel Takip ve Haritalama (PTH, Parallel Tracking and Mapping, PTAM) (Klein ve Murray 2007), PTH üzerinden geliştirilen ve Oriented FAST and Rotated BRIEF (ORB) özniteliklerinin kullanılması (Liang vd. 2016) ile oluşturulan ORB-SLAM (Mur-Artal vd. 2015, Mur-Artal ve Tardos 2017); doğrudan yöntemlerden olan Geniş Ölçek Doğrudan Tek Kameralı EZKH (GDTK EZKH, Large Scale Direct Monocular SLAM – LSD SLAM) (Engel vd. 2014), KGO (Kısmi Doğrudan Görsel Odometri, Semi-direct Visual Odometry - SVO) (Forster vd. 2014), DSO ve DDSO gibi yöntemler sayılabilmektedir.

Çizelge 2.1 Görsel EZKH Algoritmaları (Liang vd. 2016)

Sistem	Yöntem	Harita Yoğunluğu	Döngü Kapama
PTH	Öznitelik	Seyrek	Hayır
KGO	Yarı-doğrudan	Seyrek	Hayır
GDTK EZKH	Doğrudan	Yoğun	Evet
ORB SLAM	Öznitelik	Seyrek	Evet
DSO	Doğrudan	Yoğun	Hayır
DDSO	Doğrudan	Yoğun	Evet

Açık kaynaklarda bulunan EZKH algoritmalarının birçok yönden kıyaslanması ihtiyacı bulunmaktadır. Bu kıyaslamanın gerçekleştirilmesi için Filipenko ve Afanasyev (2018)'de olduğu üzere mobil bir robot kullanılabildiği gibi, masaüstü uygulamalarda kullanılabilecek veri kümeleri ile çalışmaların gerçekleştirilmesi de mümkündür.

EZKH probleminin çözümü için geliştirilen yöntemlerin çeşitliliğinin yanında, bu yöntemlerin performans analizleri için kullanılan veri kümeleri de çeşitlilik göstermektedir. Bu veri kümeleri genellikle hareket yakalama (Sturm vd. 2012) gibi yöntemler ile laboratuvarlarda oluşturulmakta olup, iç mekân, dış mekân, hareketli nesneler içeren ya da görece durağan ortamlar gibi özelliklerle birbirlerinden ayrılmaktadır.

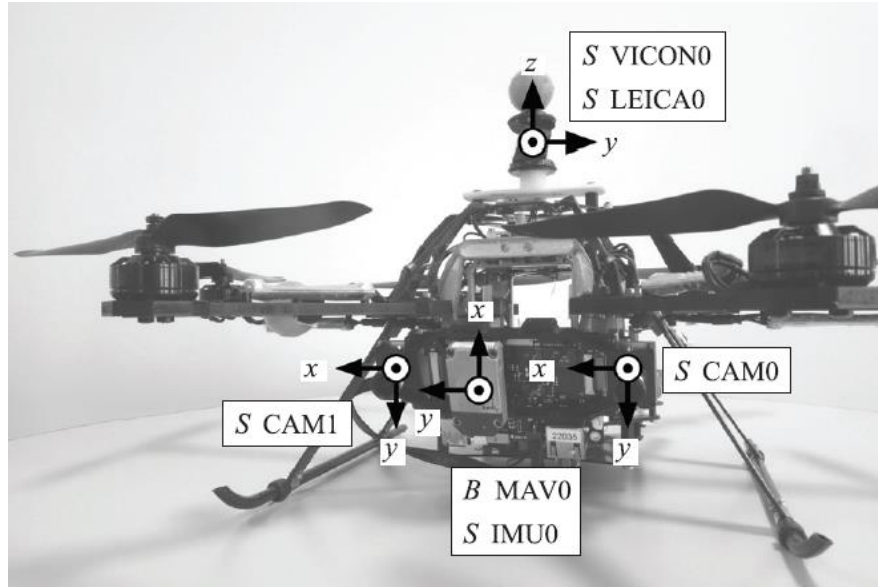
Veri kümelerinin başlıcaları arasında KITTI, EUROC, TUM KYM-D (Sturm vd. 2012), TUM Tek Kameralı Görsel Odometri (TUM TGO) veri kümesi, ICL-NUIM (Handa vd. 2014), Sintel (Butler vd. 2012), Tsukuba (Peris vd. 2012) ve NYU (Silberman vd. 2012a) veri kümeleri sayılabilmektedir.

Dış mekân verileri sağlayan KITTI veri kümesi, Şekil 2.5 (cvlibs.net, 2020) ile verildiği üzere hareket eden bir platform üzerinden kaydedilmiştir. Kamera görüntülerini, lazer taramasını, yüksek doğruluğa sahip KKS ölçümlerini, AÖB ivmelerini ve gerçek yer (ground truth) verilerini içermektedir (Geiger vd. 2013). Çift kameralı yöntemlerle de kullanılabilen KITTI veri kümesi içerisinde 11 adet eğitim, 11 adet de test verisi bulunmaktadır (Visual Odometry / SLAM Evaluation, 2012).



Şekil 2.5. KITTI veri kümesini oluşturmak için kullanılan platform ve teçhizatları (Geiger vd. 2013)

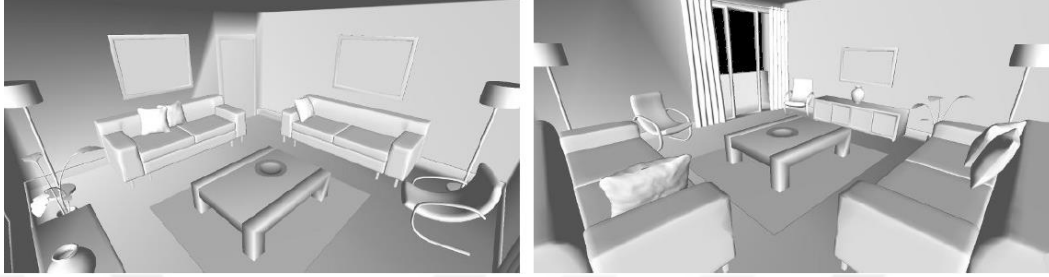
Açık kaynaklarda bulunan EUROC veri kümesi ise, Şekil 2.6 ile verildiği üzere mini bir İHA üzerinde bulunan kamera ve AÖB'ler ile toplanmaktadır. EUROC veri kümesi, kapalı mekânlarda kolay, orta ve zor olarak sınıflandırılmış 11 adet veri kümesi içerisinde, eşleştirilmiş çift kamera görüntülerini, ivme ölçümlerini ve gerçek yer verilerini içermektedir (Burri vd. 2016).



Şekil 2.6 EUROC veri toplama platformu

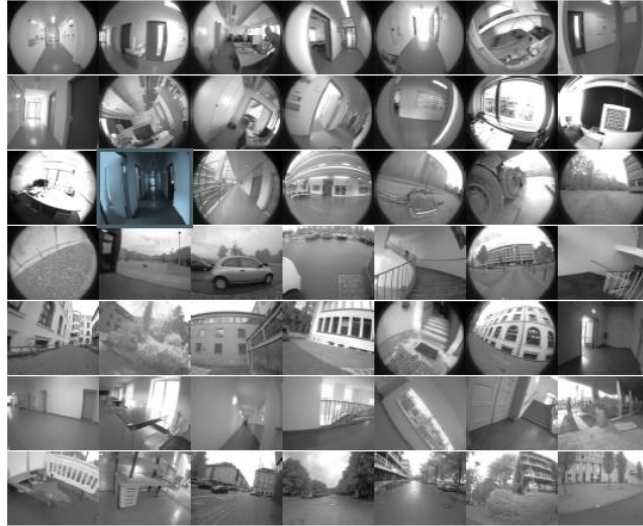
ICL-NUIM veri kümesi el kamerası hareketlerinin kullanıldığı KYM-D verileri içermektedir (Handa vd. 2014). Diğer veri kümelerinden farklı olarak ICL-NUIM veri

kümesindeki görüntüler gerçek bir ortamdan toplanmamıştır. Şekil 2.7 ile verildiği üzere ICL-NUIM veri kümesi sentetik görüntü ve derinlik bilgilerinden oluşmaktadır. officeRoom ve livingRoom olarak iki kategoriye ayrılmış olan ofis ortamının ve oturma odasının benzetimlerini içeren toplam 8 adet video dizisi ve gerçek yer verileri sunulmaktadır.



Şekil 2.7 ICL-NUIM livingRoom0 video dizisine ait farklı açılardan ortam geometrisi (Handa vd. 2014)

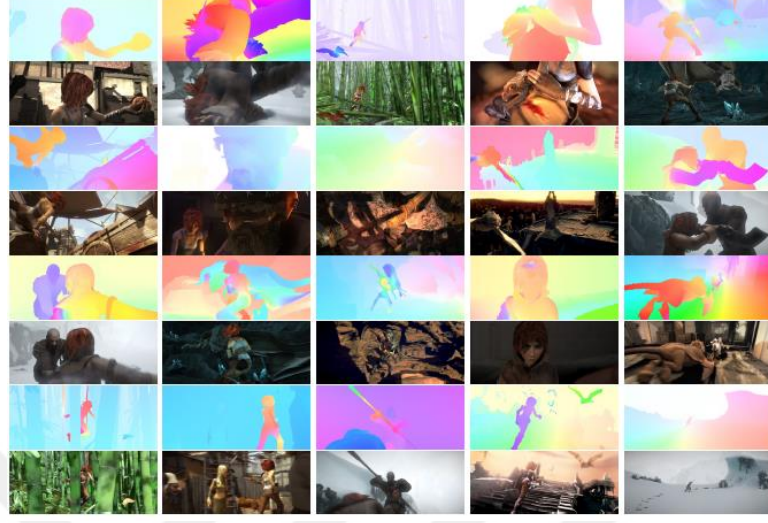
TUM TGO veri kümesi ise EZKH ve odometri yöntemlerinin değerlendirilmesinde kullanılmak üzere balıkgözü (fisheye) kamera ile toplanmış video çerçevelerinden oluşmaktadır. TUM TGO veri kümesi, dar iç ortam koridorlarından geniş dış ortamlara kadar birçok alanda toplanmış 50 farklı video dizisi içermektedir. Şekil 2.8 ile video çerçevelerinden örnekler verilmektedir.



Şekil 2.8 TUM TGO video dizilerine ait video çerçeve örnekleri (Engel vd. 2016)

Sintel isimli kısa animasyon filminden elde edilen MPI-Sintel veri kümesi hareket, ışıklandırma, sahne yapısı, materyal özelliği, atmosfer etkisi ve bulanıklık gibi oldukça

zengin etmenleri içermektedir. MPI-Sintel veri kümesinde bulunan video çerçeve örnekleri Şekil 2.9 ile verilmektedir.



Şekil 2.9 Sintel veri kümesine ait video çerçeve örnekleri (Buttler vd. 2012)

Şekil 2.10 ile bazı video çerçeveleri verilen Tsukuba veri kümesi çift kamera ile elde edilen video çerçeveleri ve yüksek doğruluğa sahip gerçek yer verisi içermektedir (Silberman vd. 2012a).



Şekil 2.10 Tsukuba veri kümesi (Silberman vd. 2012a)

NYU-Depth veri kümesi ise özellikle tipik ev ortamını betimleyen video çerçevelerinden oluşmaktadır (Silberman vd. 2012b).



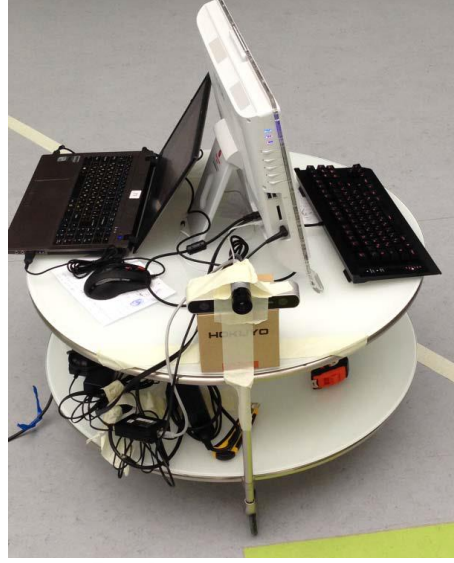
Şekil 2.11 NYU-Depth veri kümesi ev ortamı

Literatürdeki veri kümeleri ve Görsel-EZKH algoritmaları kullanılarak birçok performans karşılaştırması çalışması bulunmaktadır.

Huletski vd. (2015) ORB-SLAM, GDTK-EZKH, L-SLAM (Zikos ve Pedritis 2014) ve OpenRat-SLAM (Ball vd. 2013) yöntemlerini Intel mimarisinde işlemciye sahip bir bilgisayar kullanarak TUM KYM-D veri kümesi üzerinde karşılaştırmıştır.

Buyval vd. (2017), Robotik İşletim Sistemi (RİS, Robotic Operating System, ROS,) tabanlı ORB-SLAM, Düzenlenmiş Tek Kameralı Derinlik Kestirimi (DTKDK - Regularized Monocular Depth Estimation, REMODE) (Pizzoli vd. 2014), GDTK EZKH ve Yoğun Parçalı Düzlemsel Takip ve Haritalama (YPPTH, Dense Piecewise Planar Tracking and Mapping, DPPTAM,) (Concha ve Civera, 2015) yöntemlerinin iç ortamdaki performanslarını analiz etmiştir. Uzaktan kontrol edilen bir robot üzerinde bulunan Intel mimariye sahip bir bilgisayar ile yapılan çalışmalarda EZKH çözümlerinin harita oluşturma yetenekleri karşılaştırılmıştır.

İbragimov ve Afanasyev (2017) ise, tek kameralı ORB-SLAM ve YPPTH ile çift kameralı ZedFU ve RTABMap yöntemlerinin iç ortamdaki karşılaştırmalarını Şekil 2.12 ile verilen mobil robot üzerinde bulunan Intel işlemciye sahip bilgisayar ve kameralar ile gerçekleştirmiştir.



Şekil 2.12 Tek kameralı ORB-SLAM ve DPPTAM ile çift kameralı ZedFU ve RTAB-Map yöntemleri analiz robotu (İbragimov ve Afanasyev 2017)

Delmerico ve Scaramuzza (2018), Görsel Atalatsel EZKH yöntemlerinden olan Çoklu Durum Kısıtlaması Kalman Filtresi (ÇDKKF, Multi-State Constraint Kalman Filter, MSCKF) (Mourikis ve Roumeliotis, 2007), Açık Ana Kare Tabanlı Görsel Ataletsel EZKH (ATAGA, Open Keyframe Base Visual Inertial SLAM, OKVIS) (Leutenegger vd. 2015), Gürbüz Görsel Ataletsel Odometri (GGAO, Robust Visual Inertial Odometry, ROVIO) (Bloesch vd. 2015), VINS-Mono (Qin vd. 2018), KGO (Forster vd. 2014) +ÇDF (Çoklu Duyarga Füzyonu - Multiple Sensor Fusion, MSF) (Lynen vd. 2013) ve KGO+GTSAM (Forster vd. 2016) için EUROC veri kümesi üzerinde, MİB kullanımı, REB kullanımı ve işlem süresi gibi parametreleri karşılaştırmıştır. İlgili çalışmalar Intel ve ARM mimariye sahip dört farklı donanım üzerinde gerçekleştirilmiştir.

Filipenko ve Afanasyev, 2018 tarafından tekerlekli bir robot üzerinde RİS tabanlı tek kameralı (monocular) EZKH yöntemlerini MYH parametresi üzerinden karşılaştırmıştır. Robot üzerinde ise Ubuntu 16.04 versiyonuna sahip NVIDIA Jetson TX1 donanımı kullanılmıştır. Tek kameralı yöntemlerden GDTK EZKH, ORB-SLAM ve DSO yöntemleri karşılaştırılmıştır. Çift kameralı yöntemlerden ise Gerçek Zamanlı Görüntü Tabanlı Haritalama (GGTH, Real-Time Appearance-Based Mapping, RTAB map) (Labb'e ve Michaud, 2014), ORB SLAM, Çift Kameralı Paralel Takip ve Haritalama (ÇPTH, Stereo Parallel Tracking and Mapping, SPTAM) (Pire vd. 2017) yöntemleri karşılaştırılmıştır.



Şekil 2.13 Filipenko ve Afansayev EZKH robotu

Mingachev vd. (2020) DSO, DDSO, ORB-SLAM2 ve DynaSLAM yöntemlerini karşılaştırmıştır. Analiz kapsamında Intel mimarisinde işlemci kullanılmıştır. EUROC ve TUM veri kümeleri üzerindeki MYH ve Bağıl Durum Hatası (BDH - Relative Pose Error, RPE) parametreleri gözlenmiştir.

Merzlyakov ve Macenski (2021), ORB-SLAM3 (Campos vd. 2021), OpenVSLAM (Sumikura vd. 2019) ve RTABMAP yöntemlerini KITTI, EUROC ve TUM KYM-D veri kümeleri üzerinde inceleyerek Mutlak Yörünge Hatası açısından yöntem performansları karşılaştırmıştır.

Bu tez çalışması kapsamında ise diğer çalışmalardan farklı olarak REB, MİB ve CPU yüküne ek olarak güç tüketimi gibi donanım kısıtları ile ICL-NUIM, KITTI, EUROC ve TUM TGO gibi farklı türde veri kümeleri kullanılarak EZKH yöntemlerinin performanslarına farklı bir bakış açısı sunulmaktadır.

3. MATERYAL VE YÖNTEM

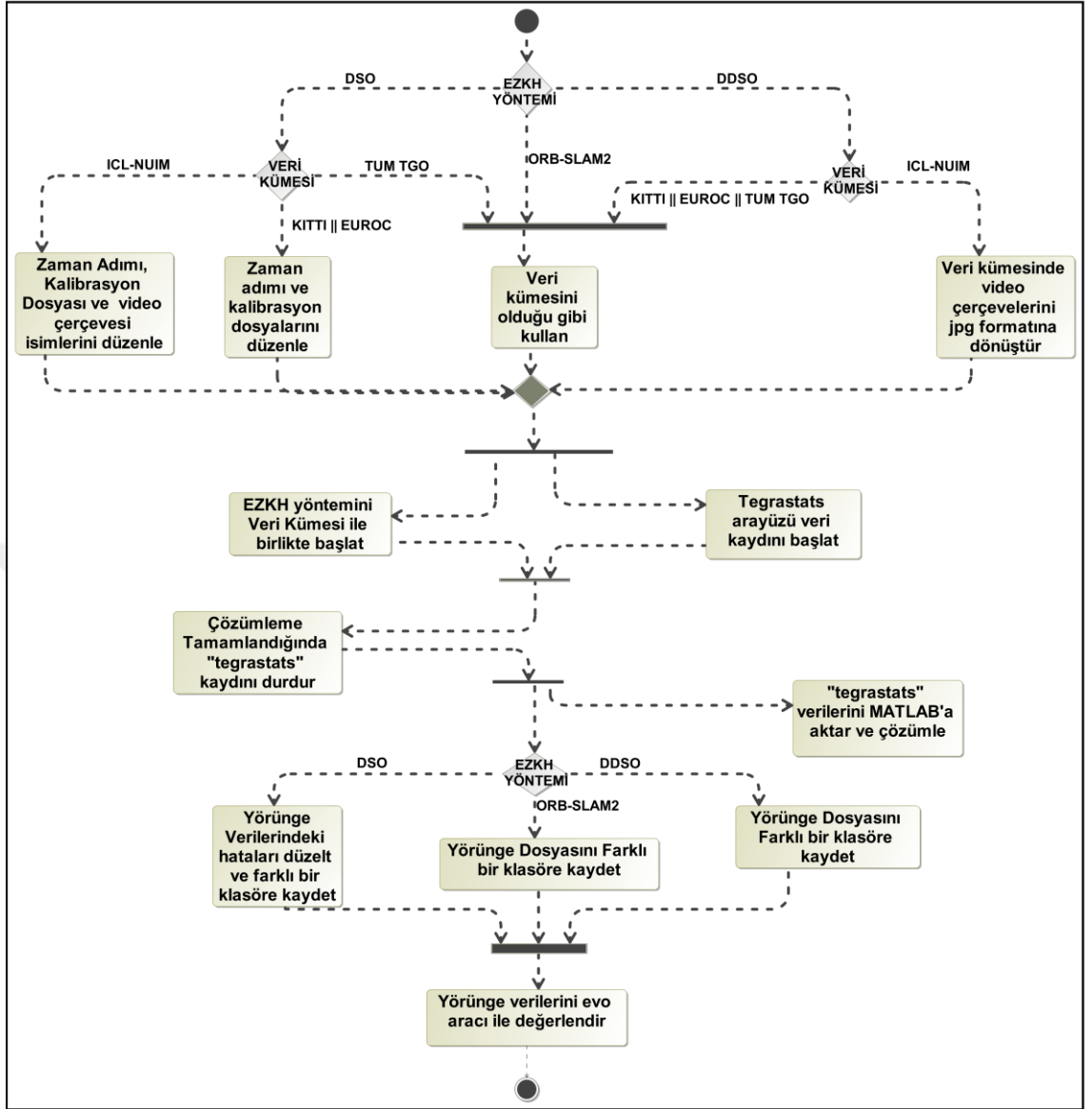
Tez kapsamında yürütülen çalışmalarda DSO, DDSO ve ORB-SLAM2 yöntemlerinin EUROCC, KITTI, ICL-NUIM ve TUM TGO veri kümelerinden alınan toplam 11 video dizisi üzerindeki davranışları ve performansları karşılaştırılmıştır. İlgili çalışmalar NVIDIA Jetson TX1 donanımı kullanılarak yürütülmüştür.

Veri kümelerinin ilgili EZKH yöntemi tarafından kullanılabilmesi için, veriye ait kalibrasyon ihtiyacı, veri kümesi içerisindeki her bir görüntünün zaman adımının tanımlanması, bu zaman adımlarını ilgili EZKH yönteminin okuma formatı, zaman adımları ile uyumlu çıktı alınabilmesi için veri kümesi içerisindeki görüntülerin doğru isimlendirilmesi, ilgili görüntünün format düzenlemesi gerçekleştirilmiştir. Bununla birlikte her bir yöntemin elde ettiği yörünge sonucunun gerçek yer verisi (ground-truth) ile kıyas edilebilmesi için gerçek yer verisi ve elde edilen yörünge verisi birbirleri ile uyumlu yapıya getirilmiştir.

Veri kümelerinin gerçek yer verisi ile karşılaştırılabilmesi için, EZKH yönteminin nasıl bir formatta rota çıktısı ürettiği, üretilen çıktıların zaman sırasına göre olup olmadığı, sonuca ilişkin formatın kıyaslamaya uygun olup olmadığı, yine ilgili dosyanın maddi hatalar içerip içermediği gibi hususlar tespit edilerek giderilmiş ve tüm veri kümelerine ilişkin alınan rota bilgileri gerçek yer verileri ile kıyaslanabilmiştir.

Açık kaynaklarda yer alan evo (MichaelGrupp, <https://github.com/MichaelGrupp/evo>, 2021) aracı ile yörünge hatasının tespit edilebilmesi için ATE parametresi kullanılmıştır.

Gerçekleştirilen çalışmalar için süreci özetleyen diyagram Şekil 3.1 ile verilmektedir.



Şekil 3.1 Çalışma Süreç Diyagramı

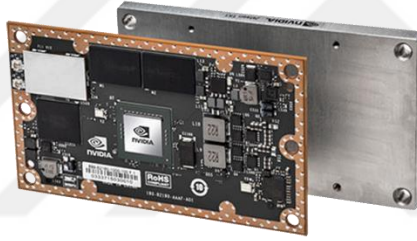
Sonraki alt bölümlerde kullanılan donanımın parametreleri, karşılaştırılan EZKH yöntemleri ve veri kümelerine ilişkin bilgiler verilmiştir.

3.1 Donanım Parametreleri

Analiz çalışmaları Çizelge 3.1 ile verilen özelliklere sahip olan NVIDIA Jetson TX1 donanımı (Şekil 3.2) üzerinde gerçekleştirilmiştir.

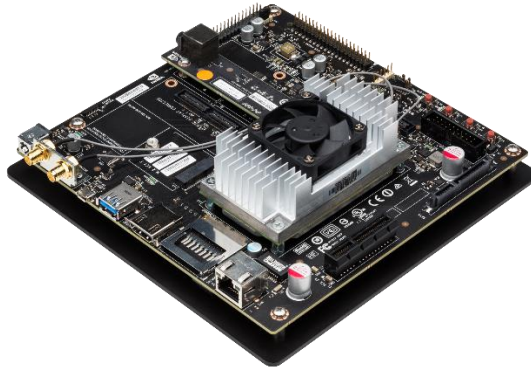
Çizelge 3.1 NVIDIA Jetson TX1 Donanım Özellikleri

Grafik İşlem Birimi	NVIDIA Maxwell™ GPU with 256 NVIDIA® CUDA® Cores
Merkezi İşlem Birimi	Quad-core ARM® Cortex®-A57 MPCore Processor
REB	4 GB LPDDR4 Memory
İşletim Sistemi	Ubuntu 18.04



Şekil 3.2 NVIDIA Jetson TX1

NVIDIA Jetson TX1 donanımı ARM mimaride işlemciye sahip bir bilgisayardır. Bununla birlikte masaüstünde doğrudan kullanılabilmesi için Şekil 3.3 ile verilen ve üzerinde HDMI, Ethernet, USB, Hafıza Kartı gibi çeşitli çevresel arayüzlerin bulunduğu geliştirme kiti bulunmaktadır.



Şekil 3.3 NVIDIA Jetson TX1 Geliştirme Kiti

Yöntemlerin işlemci üzerindeki güç tüketiminin belirlenebilmesi için NVIDIA Jetson TX1 tarafından sağlanan “tegrastats” yazılım arayüzü kullanılmıştır. Şekil 3.4’te verildiği üzere, bu arayüz, arzu edilen periyotta REB kullanımı, MİB kullanımı, GİB gücü, MİB gücü ve giriş gücü gibi detaylı bilgileri içeren bir metin dosyası sunmaktadır. Tez çalışması kapsamında 1 Hz frekansında elde edilen tegrastats verileri MATLAB aracılığıyla .mat dosyasına dönüştürülmüş ve nihai analizler MATLAB üzerinde gerçekleştirilmiştir.

```
RAM 1429/3963MB (1fb 52x4MB) SWAP 2/3005MB (cached 0MB) CPU [24%  
612,26%612,17%612,17%612] EMC_FREQ 0% GR3D_FREQ 18% PLL@27.5C  
CPU@27.5C Tboard@32C Tdiode@32.75C GPU@28C AO@33C thermal@27.75C  
PMIC@100C VDD_IN 2907/2907 VDD_GPU 229/229 VDD_CPU 688/688  
RAM 1436/3963MB (1fb 52x4MB) SWAP 2/3005MB (cached 0MB) CPU [12%  
204,21%204,3%204,7%204] EMC_FREQ 0% GR3D_FREQ 0% PLL@27C CPU@  
27.5C Tboard@32C Tdiode@32.5C GPU@28C AO@33C thermal@27.75C PMIC@  
100C VDD_IN 1874/2390 VDD_GPU 19/124 VDD_CPU 229/458  
RAM 1458/3963MB (1fb 52x4MB) SWAP 2/3005MB (cached 0MB) CPU [44%  
1734,83%1734,31%1734,48%1734] EMC_FREQ 0% GR3D_FREQ 33% PLL@  
29C CPU@31C Tboard@32C Tdiode@33.5C GPU@28.5C AO@33.5C thermal@  
29.5C PMIC@100C VDD_IN 5618/3466 VDD_GPU 152/133 VDD_CPU  
2752/1223  
RAM 1458/3963MB (1fb 52x4MB) SWAP 2/3005MB (cached 0MB) CPU [24%  
1734,42%1734,77%1734,19%1734] EMC_FREQ 0% GR3D_FREQ 48% PLL@  
29C CPU@29C Tboard@32C Tdiode@33.25C GPU@29C AO@33.5C thermal@  
29.25C PMIC@100C VDD_IN 4930/3832 VDD_GPU 152/138 VDD_CPU  
2141/1452  
RAM 1458/3963MB (1fb 52x4MB) SWAP 2/3005MB (cached 0MB) CPU [39%  
1734,34%1734,57%1734,13%1734] EMC_FREQ 0% GR3D_FREQ 33% PLL@  
29C CPU@29.5C Tboard@32C Tdiode@33.25C GPU@29C AO@34C thermal@  
29.25C PMIC@100C VDD_IN 5045/4074 VDD_GPU 229/156 VDD_CPU  
2141/1590  
RAM 1458/3963MB (1fb 52x4MB) SWAP 2/3005MB (cached 0MB) CPU [14%  
1734,38%1734,52%1734,40%1734] EMC_FREQ 0% GR3D_FREQ 31% PLL@  
29.5C CPU@29C Tboard@32C Tdiode@33.5C GPU@28.5C AO@34C thermal@  
29.25C PMIC@100C VDD_IN 5083/4242 VDD_GPU 305/181 VDD_CPU  
2140/1681  
RAM 1458/3963MB (1fb 52x4MB) SWAP 2/3005MB (cached 0MB) CPU [58%  
1734,83%1734,31%1734,48%1734] EMC_FREQ 0% GR3D_FREQ 33% PLL@  
29C CPU@31C Tboard@32C Tdiode@33.5C GPU@28.5C AO@33.5C thermal@  
29.5C PMIC@100C VDD_IN 5618/3466 VDD_GPU 152/133 VDD_CPU  
2752/1223
```

Şekil 3.4 Tegrastats Metin Dosyası

Şekil 3.5’de MİB, GİB ve giriş güçlerine ilişkin “tegrastats” arayüzü tanımı verilmiştir. Verilen tanım uyarınca, ortalama güç tüketimi değeri (Y) analiz çalışmalarında kullanılmıştır.

[VDD_name] X/Y	Current power consumption in milliwatts. X	Average power consumption in milliwatts. Y	N/A
----------------	--	--	-----

Şekil 3.5 Tegrastats Güç Verisi Tanımı

NVIDIA Jetson TX1, tegrastats arayüzü aracılığıyla REB kullanımı, MİB kullanımı ve güç tüketimi değerlerini sağlamaktadır. Bu sayede tegrastats arayüzü üzerinden donanım parametreleri kaydedilerek, EZKH yöntemlerinin çözümleme yapması esnasındaki performans parametreleri elde edilmiştir. Grafiklerde görülen sonuçların kok (kök-ortalama-kare, Root-Mean-Square, rms) değerleri alınarak, analiz çalışmalarında tüm yöntem ve veri kümeleri için kok değerleri kullanılmıştır.

3.2 Karşılaştırılan EZKH Yöntemleri

Tez çalışması kapsamında açık kaynaklarda kodları bulunan ve ARM mimaride bir işlemci üzerinde çalıştırılmaya uygun olan ORB-SLAM2, DSO ve DDSO yöntemlerinin karşılaştırılması amaçlanmıştır.

ORB-SLAM2 yöntemi tek kameralı, çift kameralı ve KYM-D kameralı çözümler üretebilmektedir. Buna karşılık DDSO ve DSO yöntemleri tek kameralı çözümler üretmektedir. Bu sebeple karşılaştırmalarda tek kameralı çözümler üzerinde durulmuştur.

Karşılaştırmalarda kullanılan ORB-SLAM2, DSO ve DDSO yöntemlerine ilişkin detaylar alt başlıklarda verilmektedir.

3.2.1 ORB-SLAM2 yöntemi

Açık kaynaklarda kaynak kodları paylaşılan öznitelik temelli ORB-SLAM2 yöntemi, tek kameralı, iki kameralı ve KYM-D duyargalar ile EZKH çözümü sağlamaktadır. En modern Görsel EZKH yöntemleri arasındadır.

ORB-SLAM2 algoritmasına ilişkin kaynak kodların bulunduğu proje sayfasında (raulmur, https://github.com/raulmur/ORB_SLAM2, 2021) belirtilen kurulum adımları uygulanarak ORB-SLAM2 algoritmasının NVIDIA Jetson TX1 donanımına kurulumu sağlanmıştır. Kurulum aşamasında MİB yükünü hafifletmek için build.sh dosyasında, Şekil 3.6’te verilen değişiklik yapılarak MİB yükü 4 MİB çekirdeğine dağıtılmıştır. Aksi takdirde kurulum işlemi tamamlanamamıştır.

```
18  cd ../..
19
20  echo "Uncompress vocabulary ..."
21
22  cd Vocabulary
23  tar -xf ORBvoc.txt.tar.gz
24  cd ..
25
26  echo "Configuring and building ORB_SLAM2 ..."
27
28  mkdir build
29  cd build
30  cmake .. -DCMAKE_BUILD_TYPE=Release
31  make -j 4
```



Şekil 3.6 ORB-SLAM2 kurulum adımları

ORB-SLAM2 algoritmasının kaynak kodu ile birlikte KITTI, EUROC ve TUM KYM-D veri kümelerinin kullanımı için hazır araçlar algoritma deposunda (repository) sunulmaktadır. Ancak ilgili veri kümelerinin doğrudan kullanılabilmesi mümkün değildir. Bu sebeple veri kümelerinde bulunan video çerçevelerinin toplandığı kameralara ait kamera kalibrasyon parametreleri de algoritma proje dosyalarında sağlanmaktadır. Algoritma içerisinde hazır olarak sağlanan veri okuma araçları da her bir veri kümesi için farklı olan bu parametreleri kullanmaktadır.

ORB-SLAM2 algoritması işlem tamamlandıktan sonra tespit ettiği yörüngeyi içeren “KeyFrameTrajectory.txt” isimli bir metin dosyası üretmektedir. Veri kümeleri ile birlikte sağlanan gerçek yer verisi ile karşılaştırmalarda “KeyFrameTrajectory.txt” isimli metin dosyası kullanılmaktadır.

3.2.2 Doğrudan Seyrek Odometri (DSO) yöntemi

Doğrudan yöntemlerden olan DSO yöntemi tek kameralı görsel EZKH algoritmasıdır. Kaynak kodların bulunduğu proje sayfasında (Jakob Engel, <https://github.com/JakobEngel/dso>, 2021) belirtilen kurulum adımları uygulanarak DSO algoritmasının NVIDIA Jetson TX1 donanımına kurulumu sağlanmıştır.

DSO algoritmasının kaynak kodu ile birlikte TUM TGO veri kümelerinin kullanımı için hazır araçlar algoritma deposunda sunulmaktadır. Farklı veri kümeleri için, veri kümelerine ait zaman adımları ve kalibrasyon verisi formatlarının DSO yöntemine uyarlanması gerekmektedir.

Ayrıca veri kümesinin tamamen okunarak tamamlanmasının ardından DSO ve DDSO yöntemleri yörüngelere (trajectory) dair zaman adımı ve konum bilgilerini içeren .txt formatında bir metin dosyası oluşturmaktadır. Yapılan çalışmalar esnasında DSO yönteminin oluşturduğu metin dosyasında sütunlar arasında fazla boşluk olduğu ve gerçek yer verisi ile karşılaştırmaya uygun olmadığı gözlenmiştir. Bu sebeple Şekil 3.7 ile verildiği üzere sütunlar arasındaki fazla boşluklar silinerek gerçek yer verisi ile karşılaştırma yapılabilmektedir.

```

0.3876814842 --1.25805137979917e-08 --1.52763142688091e-08 --4.93925389461257e-10 --1.77723610714979e-09 ·
2.14904925801424e-09 --4.57016381028618e-10 ·1¶
2.089962959¶ ·0.0452345094888488 ·0.0058058372506047 --0.00137338717941539 ·0.00287661413601977 ·
0.0823615881810968 --0.045953412031726 ·0.995538335678763¶
2.3874542713 ·0.0518629432546212 --0.00837174743462933 --0.00280039184866373 ·0.00162021291367382 ·
0.12573755811206 --0.0652541232168623 ·0.989913804729144¶
2.6541101933 ·0.0626870408832635 ·0.00996334051875 --0.00798867110091382 ·0.00246527287317681 ·
0.162958053938091 --0.0854679829671056 ·0.982921064467491¶
2.9233529568 ·0.0709679823211589 --0.0114423452551537 --0.0130047508000648 ·0.00436909523391443 ·0.200238674557674 ·
--0.106450345724444 ·0.973937219800829¶
3.187394142 ·0.0830710538234886 --0.0155546531362467 --0.0210970292572192 ·0.00503357471748096 ·0.233633749325667 ·
--0.125300923189288 ·0.964204134480613¶
3.5539836884 ·0.0929064784703512 --0.0165387474675474 --0.0279916089491221 ·0.00249629182584306 ·0.269207202954789 ·
--0.14491671592917 ·0.952113646496985¶
4.089481353 ·0.10424387651703 --0.013671656139789 ·0.03697271694605 --0.0133342512928027 ·0.298766050942225 ·
0.155100043084500 ·0.041500010100000¶

```

Şekil 3.7 DSO - Yörünge dosyasında sütunlar arası boşluklar

3.2.3 Döngü Kapatmalı Doğrudan Seyrek Odometri (DDSO) yöntemi

DDSO yöntemi, DSO algoritmasına döngü kapama özelliği eklenerek elde edilen, tek kameralı ve doğrudan EZKH çözümüdür. Kaynak kodları içeren proje sayfasında (TUM-Vision, <https://github.com/tum-vision/LDSO>, 2021) tanımlanan kurulum adımları kullanılarak NVIDIA Jetson TX1 donanımına kurulum gerçekleştirilmiştir.

Kaynak kodu ile birlikte KITTI, EUROCC ve TUM TGO veri kümeleri için hazır araçlar da sunulmaktadır.

3.3 Kullanılan Veri Kümeleri

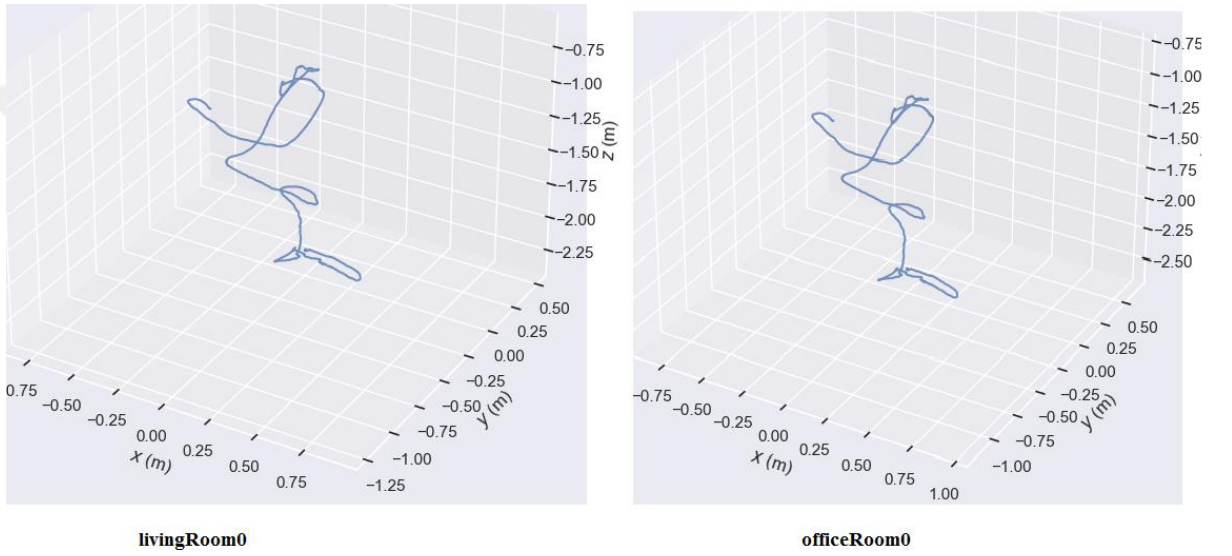
3.3.1 ICL-NUIM veri kümesi

ICL-NUIM sentetik görüntülerden oluşan bir veri kümesidir. officeRoom ve livingRoom olmak üzere iki farklı iç ortama ait saniyede 30 çerçeve hızında video dizileri sunmaktadır. Video dizileri içerisindeki her bir video çerçevesi 640x480 piksel boyutlarındadır.

Ofis ortamlarının benzetimini sunan officeRoom ve bir oturma odası benzetimi sunan livingRoom sınıflandırmaları içerisinde dört adet farklı video dizisi sunulmuştur. Hem officeRoom hem de livingRoom sınıflarında sağlanan en uzun (en fazla video çerçevesine sahip) video dizileri olan livingRoom0'ın ve officeRoom0'ın (Şekil 3.8, Şekil 3.9) analizlerde kullanılmasının uygun olacağı değerlendirilmiştir. officeRoom0 ve livingRoom0 veri kümeleri sırasıyla 1510 ve 1509 adet video çerçevesi içermektedir.



Şekil 3.8 (a) livingRoom0 video dizisine ait bir video çerçevesi , (b) officeRoom0 video dizisine ait bir video çerçevesi



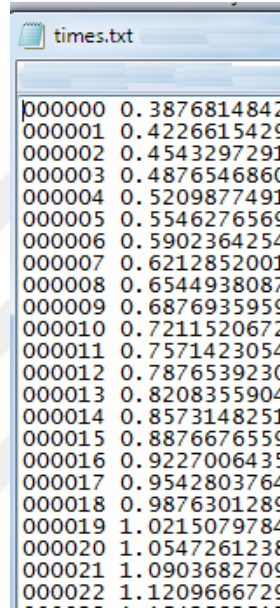
Şekil 3.9 ICL-NUIM livingRoom0 ve officeRoom0 video dizilerinin kartezyen koordinat sisteminde gösterimi

ICL-NUIM, sentetik veri kümesi sağlamakta olup yalnızca geometrik kalibrasyon ihtiyacı bulunmaktadır. Geometrik kalibrasyon için ise Denklem (2) ile verilen K kalibrasyon matrisi kullanılmıştır (Handa vd. 2014).

$$K = \begin{bmatrix} 0.751875 & 0 & 0.5 \\ 0 & -1 & 0.5 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

ICL-NUIM veri kümesi iki formatta sunulmaktadır. Bunlardan biri doğrudan TUM-KYM-D veri kümesi formatında olup, ORB-SLAM2 ile uyumlu klasör yapısı, zaman adımı tanımlaması ve görüntü formatına sahiptir.

Ancak DDSO'nun jpg formatındaki görüntüler üzerinde çalışıyor olması sebebi ile, ICL-NUIM veri kümesinin sağladığı png formatındaki görüntüler jpg formatına dönüştürülmüştür. Bununla birlikte “1, 2, 3...” şeklinde tanımlanan görüntü isimleri, DSO yönteminin sıralı okuma yöntemine uygun olarak “000001, 000002, 000003...” şeklinde düzenlenmiştir. Zaman adımları ise Şekil 3.10'da verildiği üzere “times.txt” dosyası içerisinde DSO ve DDSO'ya uygun olarak [isim zamanAdımı] şeklinde oluşturulmuştur.



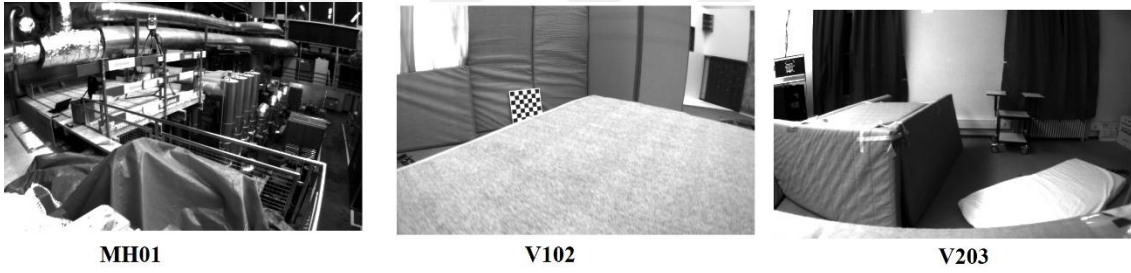
000000	0.3876814842
000001	0.4226615429
000002	0.4543297291
000003	0.4876546860
000004	0.5209877491
000005	0.5546276569
000006	0.5902364254
000007	0.6212852001
000008	0.6544938087
000009	0.6876935959
000010	0.7211520672
000011	0.7571423054
000012	0.7876539230
000013	0.8208355904
000014	0.8573148251
000015	0.8876676559
000016	0.9227006435
000017	0.9542803764
000018	0.9876301289
000019	1.0215079784
000020	1.0547261238
000021	1.0903682709
000022	1.1209666729

Şekil 3.10 “times.txt” metin dosyasındaki satır-sütun yapısı - [isim zamanAdımı]

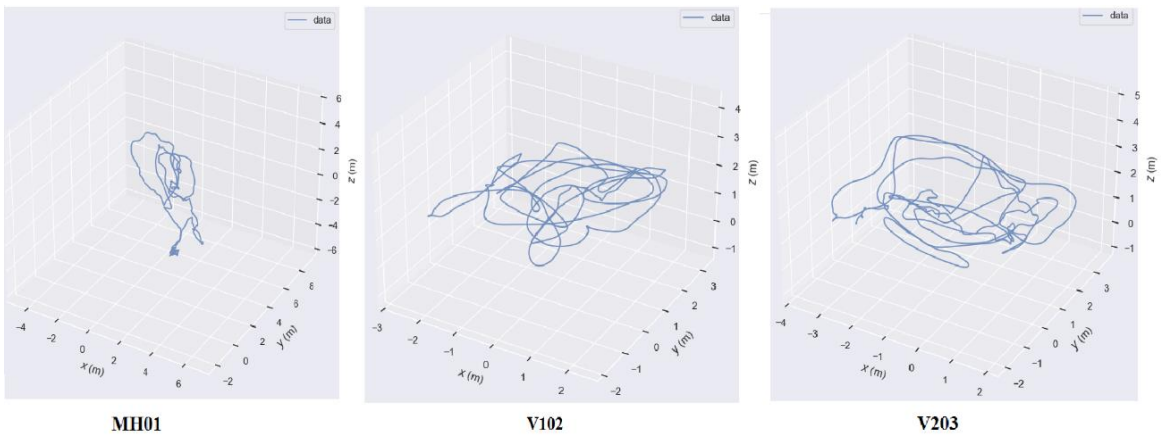
Zaman adımı ve görüntülerin bulunduğu klasör yapısı da yine DSO ve DDSO okuma yöntemlerine uygun olarak ayarlanmıştır.

3.3.2 EUROC veri kümesi

Her biri saniyede 20 çerçeve hızına sahip ve bir İHA ile toplanan EUROC veri kümesi kapsamında sağlanan MH01_Easy, V102-Medium, V203-Difficult (Şekil 3.11, Şekil 3.12) isimli 752x480 piksel çözünürlükte iç ortam görüntüleri içeren veri kümeleri üzerinde DSO, DDSO ve ORB SLAM2 yöntemleri karşılaştırılmıştır. Kolaydan zora sıralanan veri kümelerinde, bulanıklaşma, hareket etkisi, ışıklandırma, kameranın açisal hızı ve doğrusal hızı gibi değişken etkenler video dizilerinin sınıflandırılmasında kullanılan kıstaslardır. Bu sebeple karşılaştırmalarda kolay, orta ve zor sınıflarından veri kümeleri seçilerek çalışmalar gerçekleştirilmiştir. Çalışmada kullanılacak olan MH01 isimli veri kümesinde 3682 adet video çerçevesi varken, V102 isimli video dizisinde 1711 adet, V203 isimli video dizisinde ise 1922 adet 8 bit gri seviye video çerçevesi bulunmaktadır.



Şekil 3.11 EUROC veri kümesi MH01, V102 ve V203 video dizilerine ait video çerçeveleri



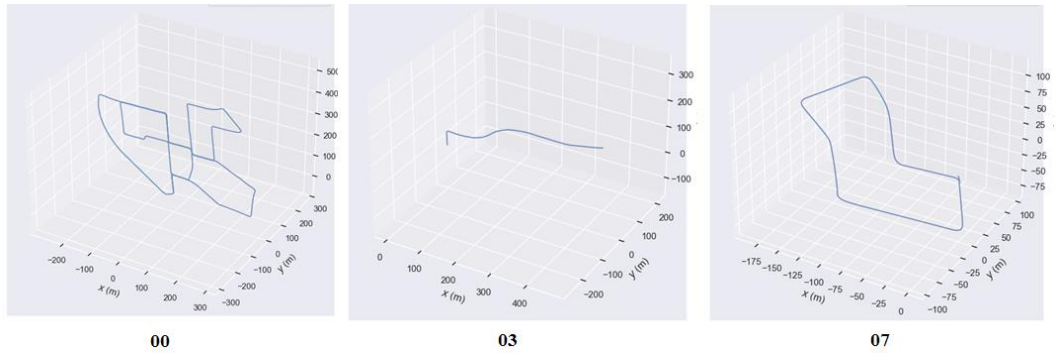
Şekil 3.12 EUROC veri kümesi MH01, V102 ve V203 isimli video dizilerinin kartezyen koordinatta gösterimi

3.3.3 KITTI veri kümesi

Her biri saniyede 10 çerçeve hızına sahip ve KITTI veri kümesi kapsamında sağlanan 00, 03 ve 07 (Şekil 3.13) isimli dış ortam görüntüleri içeren veri kümeleri üzerinde DSO, DDSO ve ORB SLAM2 yöntemleri karşılaştırılmıştır. Farklı kamera kalibrasyon matrislerine sahip olan 00, 03 ve 07 video dizilerinin (Şekil 3.13) kartezyen koordinat üzerindeki rotaları Şekil 3.14’de verilmektedir. Veri kümeleri kamera görüntülerinin yanında, lazer taramasını, yüksek doğruluğa sahip KKS ölçümlerini, AÖB ivmelerini ve gerçek yer verilerini içeren (Geiger vd. 2013) 00, 03 ve 07 isimli video dizileri sırasıyla 1246x376, 1242x375 ve 1226x370 piksel çözünürlükte 4541 adet, 801 adet ve 1101 adet 8 bit gri seviye video çerçevesi içermektedir.



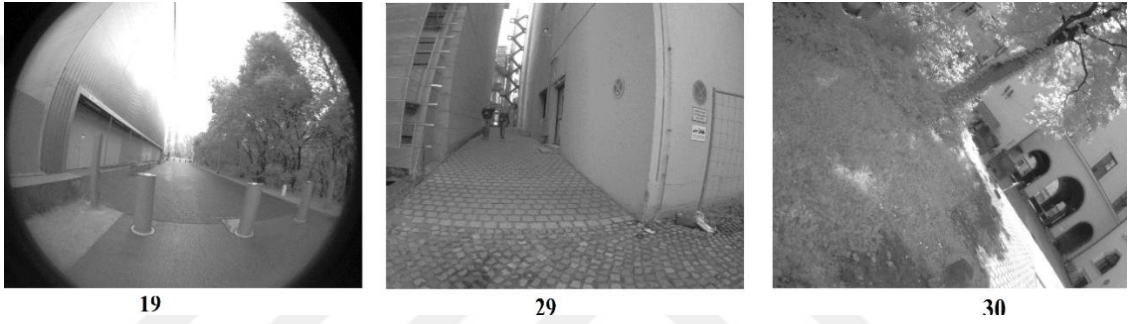
Şekil 3.13 KITTI veri kümesi 00, 03, 07 isimli video dizilerine ait video çerçevesi örnekleri



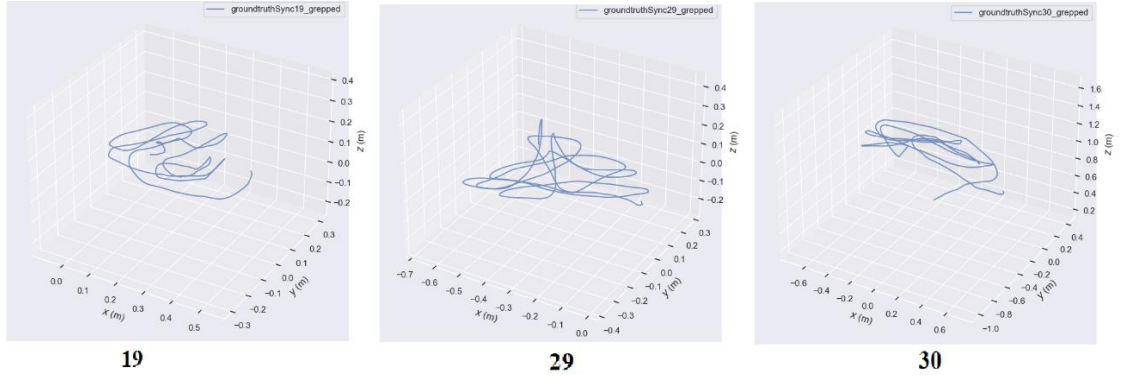
Şekil 3.14 KITTI veri kümesi 00, 03 ve 07 isimli video dizilerinin kartezyen koordinatta gösterimi

3.3.4 TUM TGO veri kümesi

Balıkgözü (fisheye) kamera ile elde edilen TUM TGO veri kümesinde sırasıyla saniyede 40, 30 ve 20 çerçeve hızına sahip olan 19, 29 ve 30 numaralı video dizileri karşılaştırmalarda kullanılmıştır. ORB-SLAM2'nin balıkgözü kameralara uygun altyapısı olmaması sebebiyle ORB-SLAM2 bu değerlendirmeye katılmamıştır. Şekil 3.15 ve Şekil 3.16 ile 19, 29 ve 30 numaralı video dizilerine ait görüntüler ve yörüngeler verilmektedir. TUM TGO 19 video dizisinde 8380 adet, TUM TGO 29 video dizisinde 8400, TUM TGO 30 video dizisinde ise 1800 adet 1280×1024 piksel çözünürlükte video çerçevesi bulunmaktadır.



Şekil 3.15 TUM TGO veri kümesi 19, 29 ve 30 isimli video dizilerine ait video çerçeveleri



Şekil 3.16 TUM TGO veri kümesi 19, 29, 30 isimli video dizilerinin Kartezyen koordinatta gösterimi

3.4 Karşılaştırma Metriği

3.4.1 Mutlak Yörünge Hatası (MYH)

DSO, DDSO ve ORB2 ile gerçekleştirilen çalışmalarda veri kümeleri üzerinde konumlandırma doğruluğu için MYH değerlendirmesi kullanılmıştır. Robotun hesaplanan konumunun gerçek yer verisinden ne kadar uzak olduğu metre cinsinden MYH ile tanımlanmaktadır.

3.4.2 Çerçeve başına çözümleme süresi

EZKH çözüm yöntemlerinin çözüm hızlarının karşılaştırılabilmesi için çerçeve başına süre bilgisi (milisaniye) deney sonuçlarında verilmektedir.

3.4.3 Rastgele Erişimli Bellek (REB) kullanımı

Donanım kısıtlarının belirlenmesi ve EZKH çözümlerinin getirdiği hafıza yükünün belirlenebilmesi için NVIDIA Jetson TX1 donanımını tegrastats arayüzüyle sağlanan (Şekil 3.4) REB kullanım bilgisi (MB) karşılaştırmalarda kullanılmıştır.

3.4.4 Merkezi İşlem Birimi (MİB) kullanımı

Donanımın hesaplama yükünün değerlendirilmesinde öncelikli olarak bakılan MİB yükü NVIDIA Jetson TX1 donanımının tegrastats arayüzü vasıtasıyla yüzdesel olarak elde edilerek karşılaştırmalarda kullanılmıştır.

3.4.5 Grafik İşlem Birimi (GİB) güç tüketimi

MİB yüküne ek olarak, GİB güç tüketiminin de toplam güce etkisini belirlemek amacıyla karşılaştırmalarda GİB güç tüketimi de miliwatt cinsinden elde edilerek kullanılmıştır.

3.4.6 Güç tüketimi

EZKH yöntemleri genellikle robotik sistemlerde kullanılmaktadır. Robotlar, kararlı ve sağlam bir hareket elde etmek için durum tahminlerinde doğruluk ve düşük gecikme oranlarının bir kombinasyonunu gerektirir. Bununla birlikte, robotların güç ve yük kısıtlamaları nedeniyle, durum tahmini algoritmaları bu nitelikleri gömülü donanımın hesaplama kısıtlamaları altında sağlamalıdır (Delmerico ve Scaramuzza, 2018). Mobil robotların güç kısıtları göz önünde bulundurularak, EZKH yöntemlerinin robotun güç

tüketimine etkisinin araştırılması amacıyla NVIDIA Jetson TX1 donanımının çözüm üretme aşamasındaki toplam güç tüketimi miliwatt cinsinden karşılaştırmalarda kullanılmıştır.



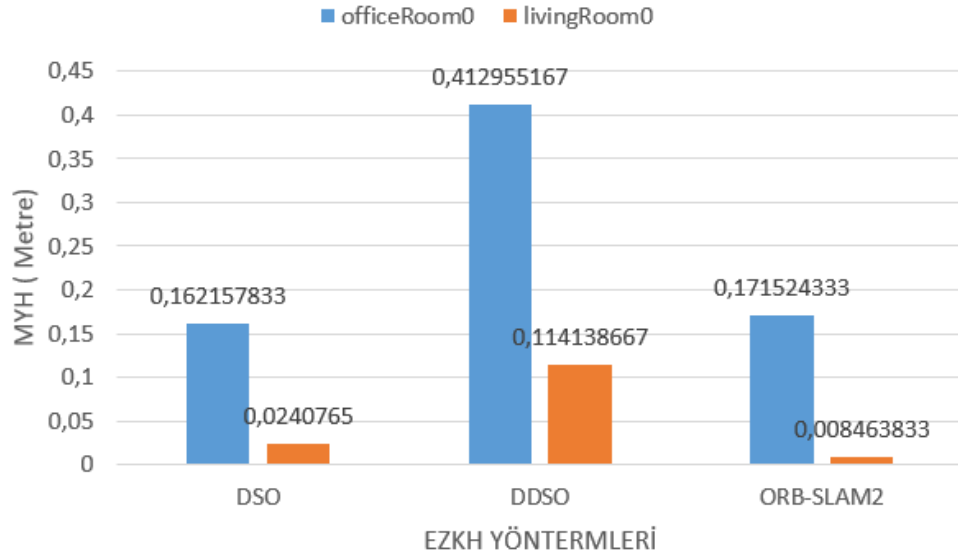
4. EZKH YÖNTEMLERİNİN KARŞILAŞTIRILMASI

Tez çalışması kapsamında DSO, DDSO ve ORB-SLAM2 yöntemlerinin ICL-NUIM, KITTI, EUROOC ve TUM TGO veri kümelerine ait video dizileri üzerindeki performansları karşılaştırılmıştır. Performans parametreleri açısından “Mutlak Yörünge Hatası”, çerçeve başına işlem süresi, REB kullanımı, MİB ve GİB yükleri ile toplam güç tüketimi parametreleri göz önüne alınmıştır. Rastsal sonuçların önüne geçebilmek için her EZKH yöntemi video dizileri üzerinde altı kez çalıştırılmış ve parametreler için elde edilen bu altı adet sonucun ortalaması karşılaştırmalarda kullanılmıştır. Devam eden başlıklarda ilgili parametreler için elde edilen sonuçların detayları verilmektedir.

4.1 Mutlak Yörünge Hatası

EZKH yöntemlerinin karşılaştırılmasında en sık başvuru parametrelerin başında MYH gelmektedir. MYH, EZKH yönteminin gerçek yer verisini ne kadar iyi takip edebildiğinin bir göstergesidir. Bu sebeple DSO, DDSO ve ORB-SLAM2 yöntemlerinin veri kümeleri üzerindeki performanslarının karşılaştırılmasında MYH parametresi göz önüne alınmıştır. MYH parametresinin hesaplanması için açık kaynaklarda bulunan “evo” aracı kullanılmış ve elde edilen yörünge dosyası gerçek yer verisi ile kıyaslanmıştır. ICL-NUIM, KITTI, EUROOC ve TUM TGO veri kümeleri için elde edilen sonuçlar müteakip paragraflarda verilmektedir.

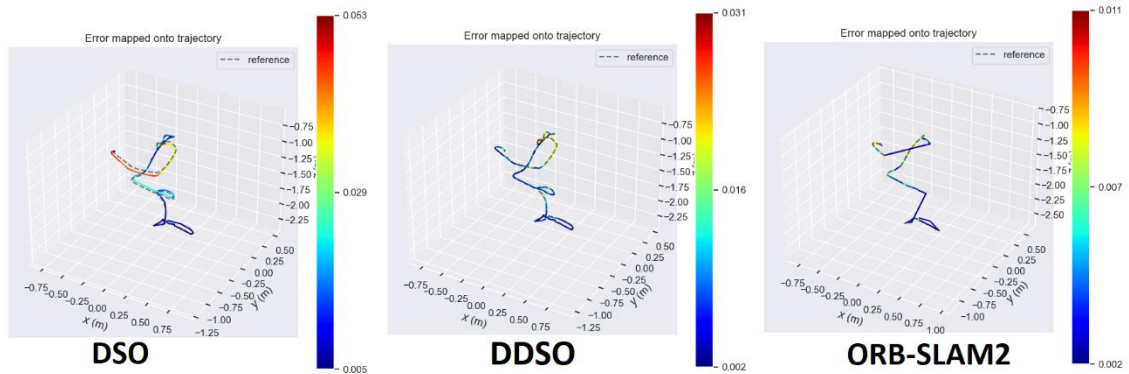
ICL-NUIM veri kümesine ait officeRoom0 ve livingRoom0 video dizileri üzerinde DSO, DDSO ve ORB-SLAM2 yöntemleri ile elde edilen MYH sonuçları Şekil 4.1 ile verilmiştir.



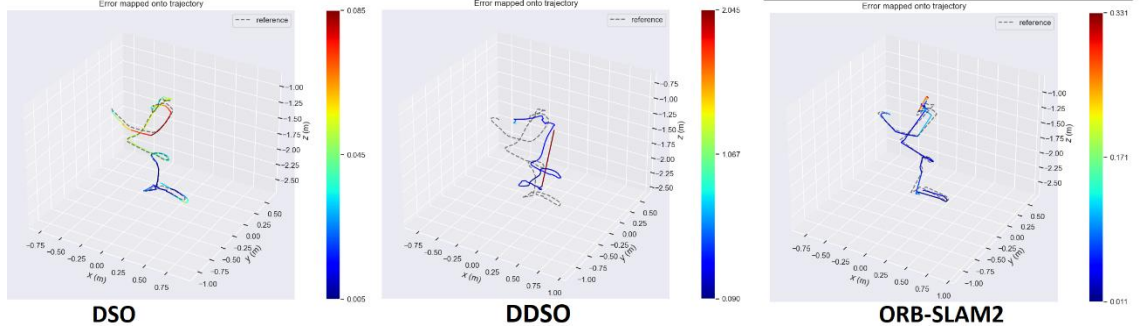
Şekil 4.1 ICL-NUIM veri kümesi üzerinde DSO, DDSO ve ORB-SLAM2 yöntemleri ile elde edilen MYH değerleri (kok)

Şekil 4.1 incelendiğinde MYH sonuçları açısından livingRoom0 video dizisi üzerinde en iyi sonucu ORB-SLAM2'nin verdiği, officeRoom0 video dizisi üzerinde ise en iyi sonucu DSO yönteminin verdiği görülmektedir. Ayrıca officeRoom0 için DDSO ve ORB-SLAM2 yöntemlerinin sonuçlarında kayda değer bir bozulma (LivingRoom0'a kıyasla sırasıyla ~30 ve ~20 kat) olduğu fark edilmektedir.

EZKH yöntemleri ile elde edilen yörüngelerin, livingRoom0 ve officeRoom0 video dizilerine ait gerçek yer verisi üzerindeki iz düşümleri Şekil 4.2 ve Şekil 4.3'te verilmektedir.

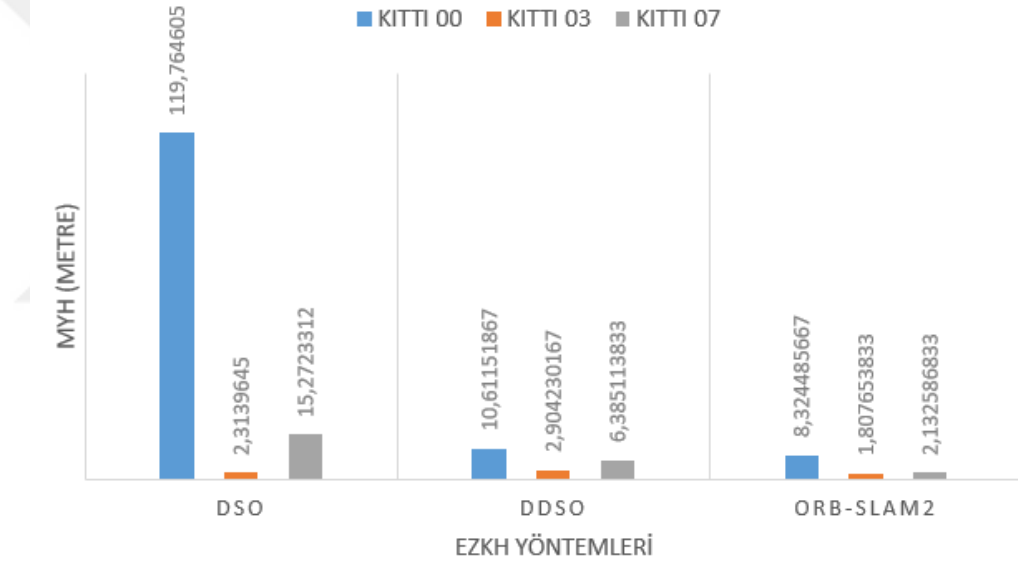


Şekil 4.2 livingRoom0 - DSO, DDSO ve ORB-SLAM2 yöntemleri ile elde edilen yörüngelerin gerçek yer verileri üzerindeki iz düşümleri



Şekil 4.3 officeRoom0 - DSO, DDSO ve ORB-SLAM2 yöntemleri ile elde edilen yörüngelerin gerçek yer verileri üzerindeki iz düşümleri

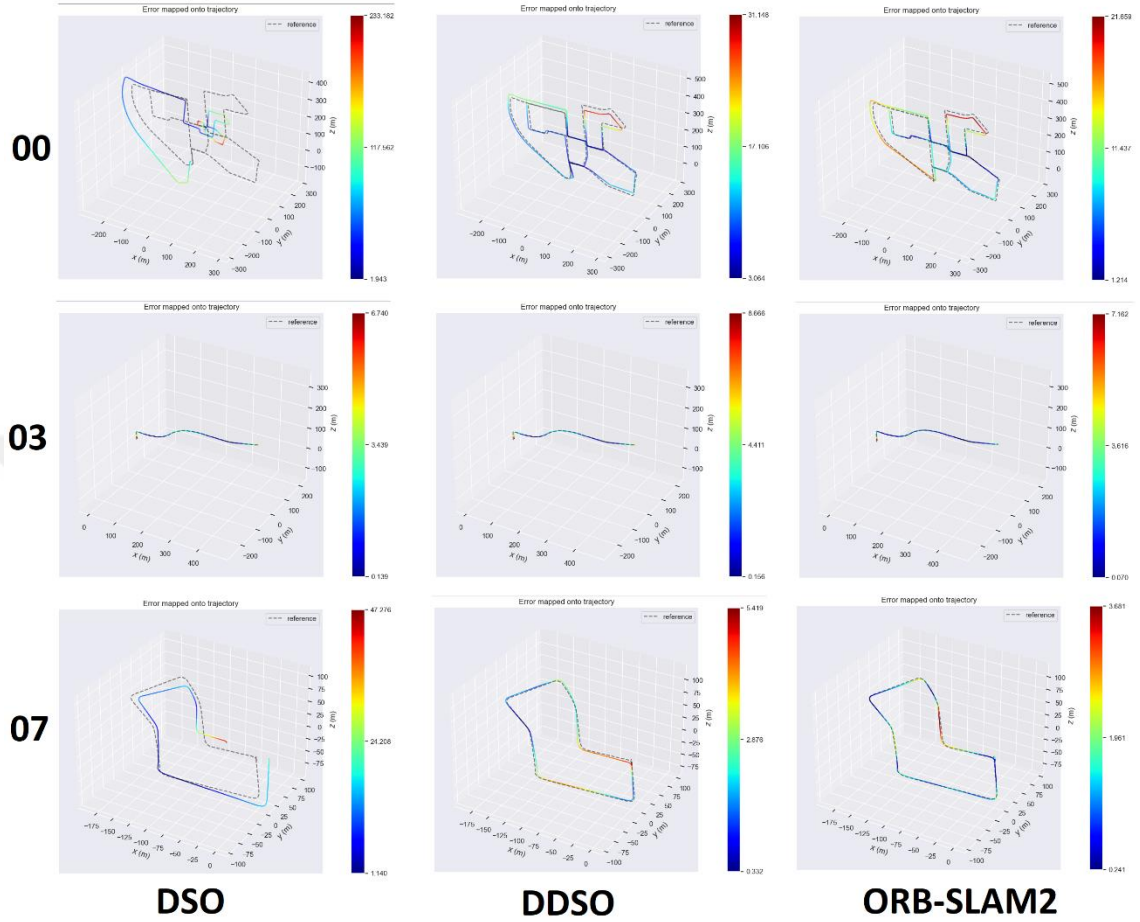
KITTI veri kümesine ait 00, 03 ve 07 isimli video dizileri üzerinde DSO, DDSO ve ORB-SLAM2 yöntemleri ile elde edilen MYH değerleri Şekil 4.4 ile verilmektedir.



Şekil 4.4 KITTI veri kümesine ait video dizileri üzerinde DSO, DDSO ve ORB-SLAM2 yöntemleri ile elde edilen MYH değerleri (kok)

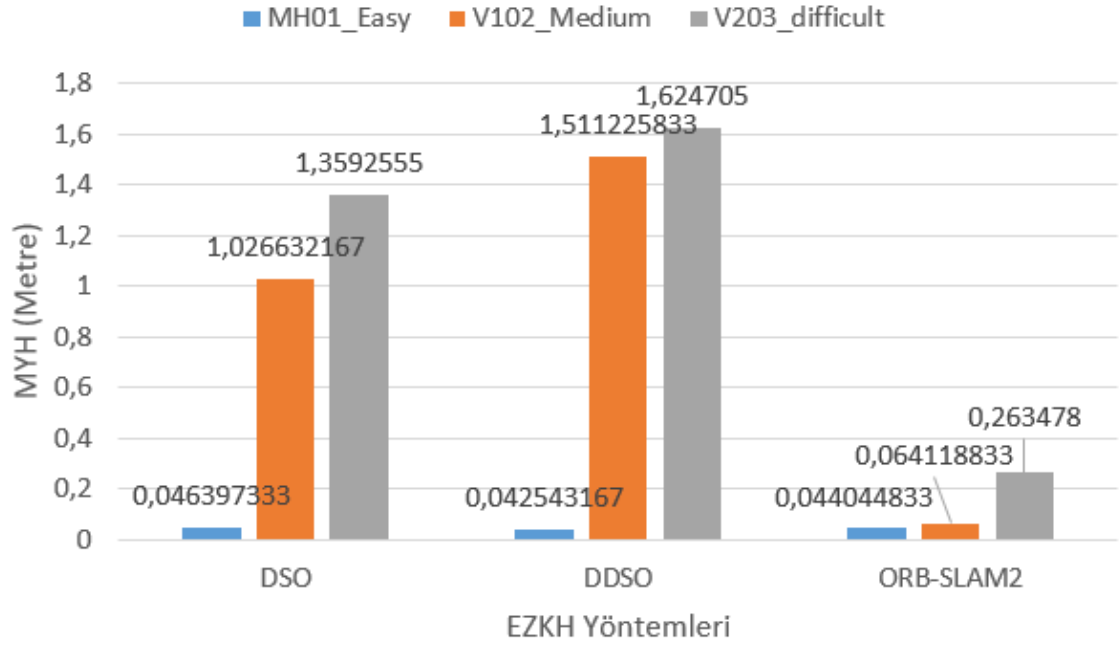
Şekil 4.4'teki sonuçlar, KITTI veri kümesine ait tüm video dizileri üzerinde en iyi sonucu ORB-SLAM2 yönteminin verdiğini göstermektedir. Bununla birlikte 03 numaralı video dizisinde olduğu gibi, dönüşlerin görece az olduğu kısa senaryolarda (Bkz. Şekil 3.14) DSO yönteminde MYH'nin dikkat çekici bir şekilde küçüldüğü ve DDSO ve ORB-SLAM2 yöntemine yakınsadığı gözlenmektedir. Diğer bir deyişle, dönüşlerin ve hareketin daha çok olduğu senaryolarda döngü kapama kabiliyetinin MYH değerini küçültme etkisi 03 numaralı video dizisi üzerinde gözlenmektedir.

DSO, DDSO ve ORB SLAM2 yöntemleri ile elde edilmiş yörüngelerin, 00, 03 ve 07 video dizilerine ait gerçek yer verisi üzerindeki iz düşümleri Şekil 4.5 ile verilmektedir.



Şekil 4.5 KITTI veri kümesine ait 00, 03, 07 isimli video dizilerinde DSO, DDSO ve ORB-SLAM2 yöntemleri ile elde edilen yörüngelerin gerçek yer verisi üzerindeki iz düşümleri

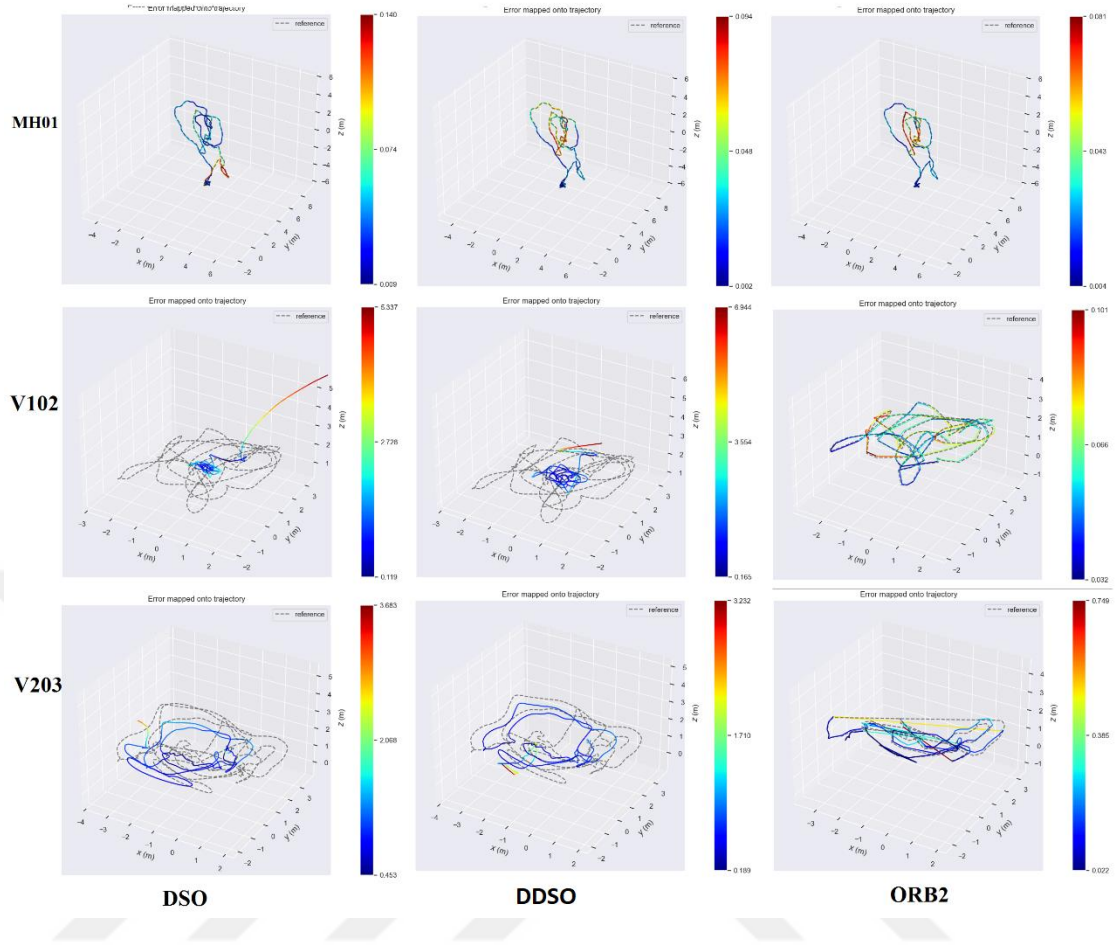
EUROC veri kümesine ait MH01, V102 ve V203 video dizileri üzerinde DSO, DDSO ve ORB-SLAM2 yöntemleri ile elde edilen MYH (kok) değerleri Şekil 4.6 ile verilmektedir.



Şekil 4.6 EUROC veri kümesine ait video dizileri üzerinde DSO, DDSO ve ORB-SLAM2 yöntemleri ile elde edilen MYH değerleri (kok)

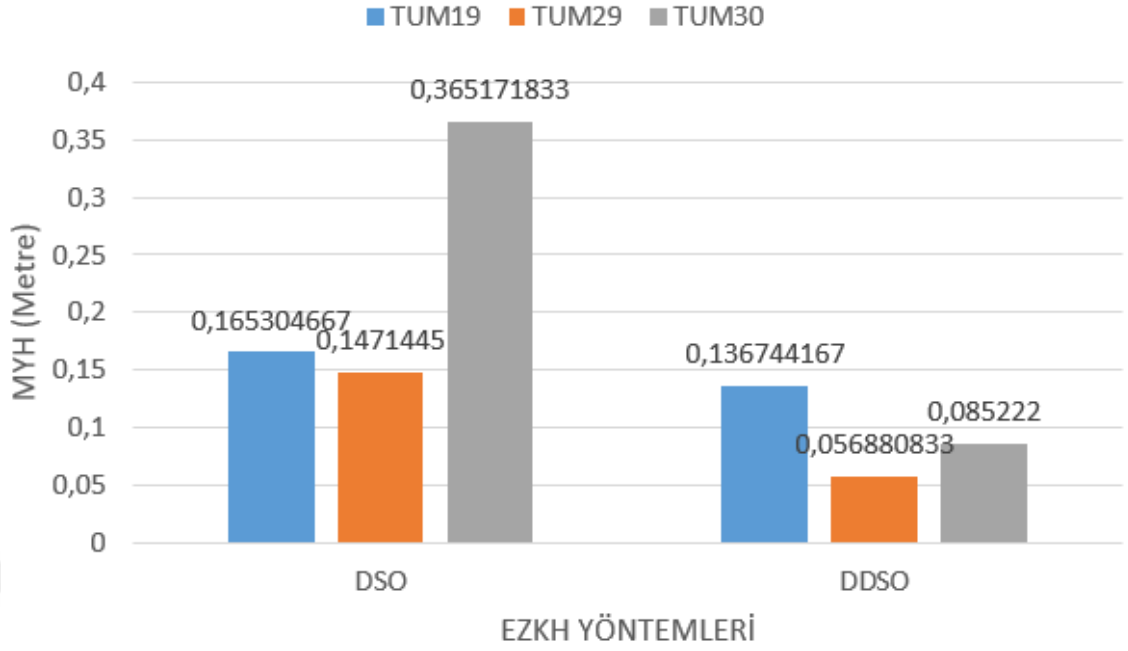
EUROC veri kümesi üzerinde elde edilen MYH değerleri incelendiğinde EUROC veri kümesi üzerinde en iyi sonucu ORB-SLAM2 yönteminin verdiği görülmektedir.

DSO, DDSO ve ORB SLA”M2 yöntemleri ile elde edilmiş yörüngelerin, MH01, V102 ve V203 veri kümelerine ait gerçek yer verisi üzerindeki iz düşümleri Şekil 4.7 ile verilmektedir.



Şekil 4.7 EUROC veri kümesine ait MH01, V102, V203 isimli video dizilerinde DSO, DDSO ve ORB-SLAM2 yöntemleri ile elde edilen yörüngelerin gerçek yer verisi üzerindeki iz düşümleri

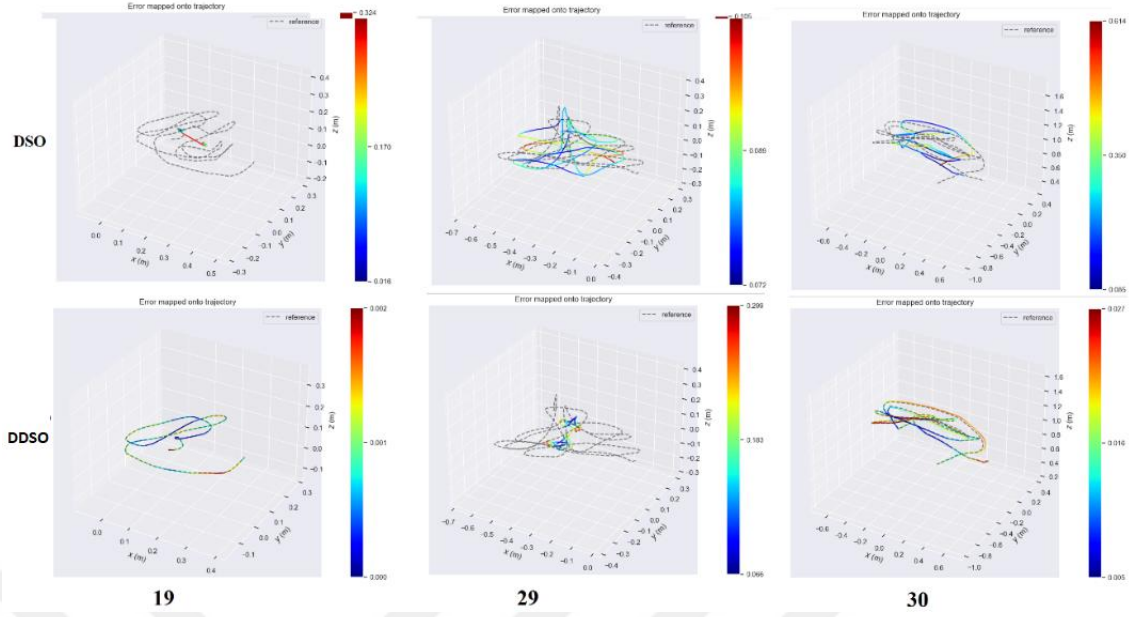
TUM TGO veri kümesine ait 19, 29 ve 30 numaralı video dizileri üzerinde DSO ve DDSO yöntemleri ile elde edilen MYH (kok) değerleri Şekil 4.8 ile verilmektedir.



Şekil 4.8 TUM TGO veri kümesine ait video dizileri üzerinde DSO ve DDSO yöntemleri ile elde edilen MYH değerleri (kok)

TUM TGO veri kümesi üzerindeki MYH sonuçları incelendiğinde TUM TGO veri kümesi üzerinde döngü kapama işleminin etkisi net bir şekilde görülmektedir. Bu haliyle DDSO yönteminin MYH değerleri DSO'ya kıyasla bir hayli düşük elde edilmiştir.

DSO ve DDSO yöntemleri ile elde edilmiş yörüngelerin, 19, 29 ve 30 numaralı video dizilerine ait gerçek-yer verisi üzerindeki iz düşümleri Şekil 4.9 ile verilmektedir.



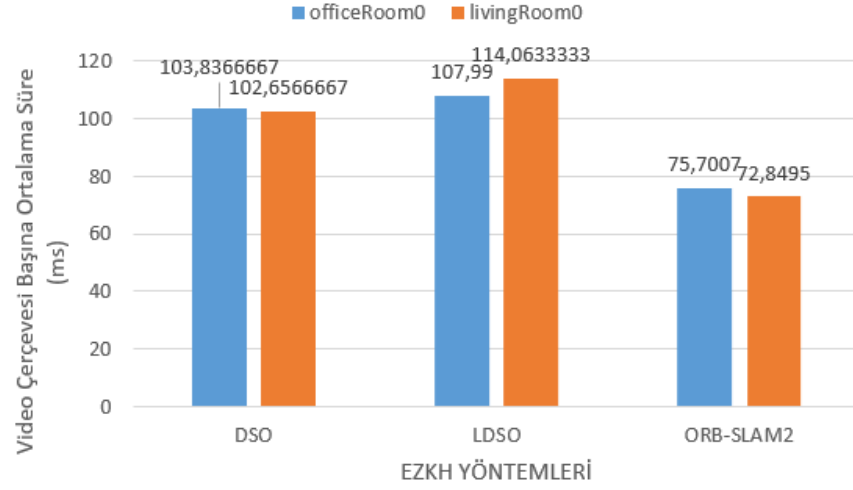
Şekil 4.9 DSO ve DDSO yöntemleri ile TUM TGO veri kümesine ait 19, 29 ve 30 numaralı video dizileri üzerinde elde edilen yörüngenin gerçek yer verisi üzerindeki iz düşümü

4.2 Çerçeve Başına İşlem Süresi

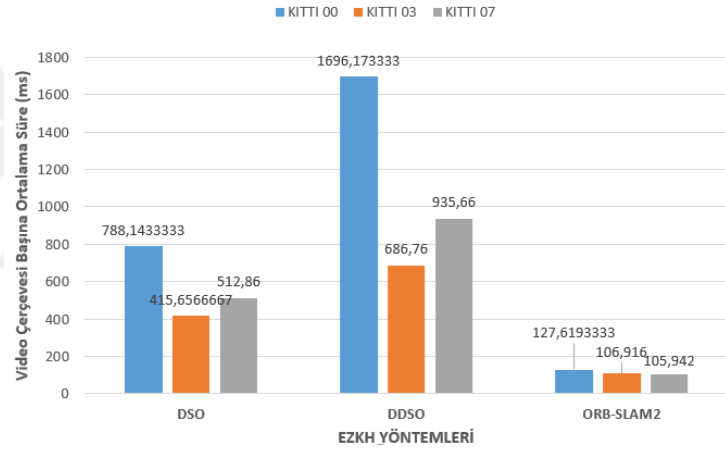
Algoritmanın başarımının yanında, bu başarımın hangi kısıtlar içerisinde gerçekleştiği de göz önünde bulundurulması gereken hususlardan bir tanesidir. Özellikle otonom sistemlerde gerçek zamanlı işlemler açısından algoritmanın çözümleme yapma hızı önem arz etmektedir. Donanım kısıtlarına da bağlı olan çözümleme hızı, maliyet, ağırlık ve güç bütçesi kısıtlarını da doğrudan etkilemektedir. Bu sebeple EZKH yöntemlerinin işlem hızı açısından karşılaştırılması amaçlanmıştır.

DSO, DDSO ve ORB-SLAM2 yöntemlerinin ICL-NUIM, KITTI, EUROCC ve TUM TGO veri kümelerine ait video dizileri üzerindeki karşılaştırmalara ilişkin sonuçlar sırasıyla Şekil 4.10, Şekil 4.11, Şekil 4.12 ve Şekil 4.13'te verilmektedir.

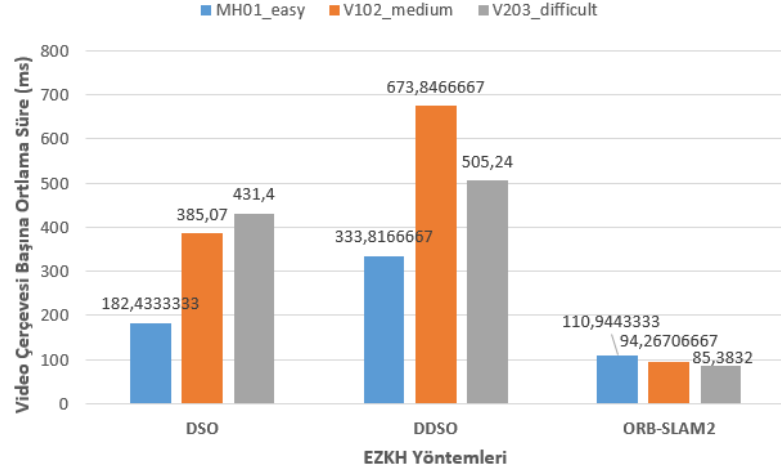
ICL-NUIM, KITTI ve EUROCC veri kümeleri üzerinde en hızlı çözümlemeyi ORB-SLAM2 yönteminin gerçekleştirdiği görülmüştür. Bununla birlikte yalnızca DSO ve DDSO yöntemlerinin karşılaştırıldığı TUM TGO veri kümesi de dâhil olmak üzere, DSO yöntemi tüm veri kümeleri üzerinde DDSO yöntemine kıyasla daha hızlı sonuç üretmektedir. Ek olarak ICL-NUIM veri kümesi üzerinde DSO ve DDSO yöntemlerinin çözümleme hızları birbirine yakın olmuştur.



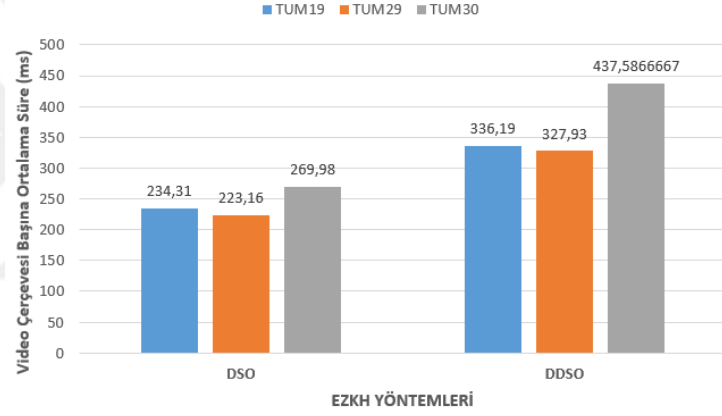
Şekil 4.10 ICL-NUIM veri kümesine ait officeRoom0 ve livingRoom0 video dizileri üzerinde EZKH yöntemlerinin çözümü süreleri (ms)



Şekil 4.11 KITTI veri kümesine ait video dizileri üzerinde EZKH yöntemlerinin çözümü süreleri (ms)



Şekil 4.12 EUROK veri kümesine ait video dizileri üzerinde EZKH yöntemlerinin çözümleme süreleri (ms)

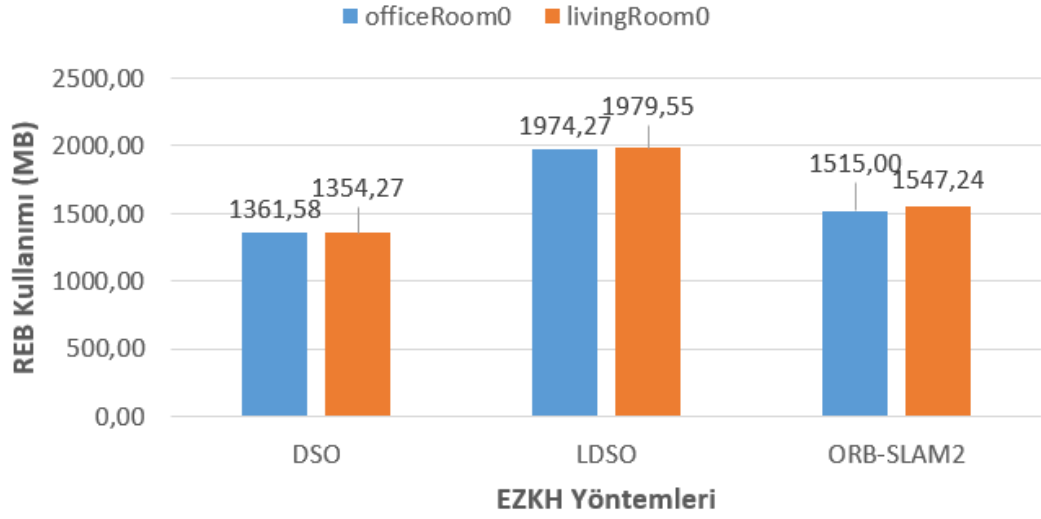


Şekil 4.13 TUM TGO veri kümesine ait video dizileri üzerinde DSO ve DDSO yöntemlerinin çözümleme süreleri (ms)

4.3 Rastgele Erişimli Bellek Kullanımı

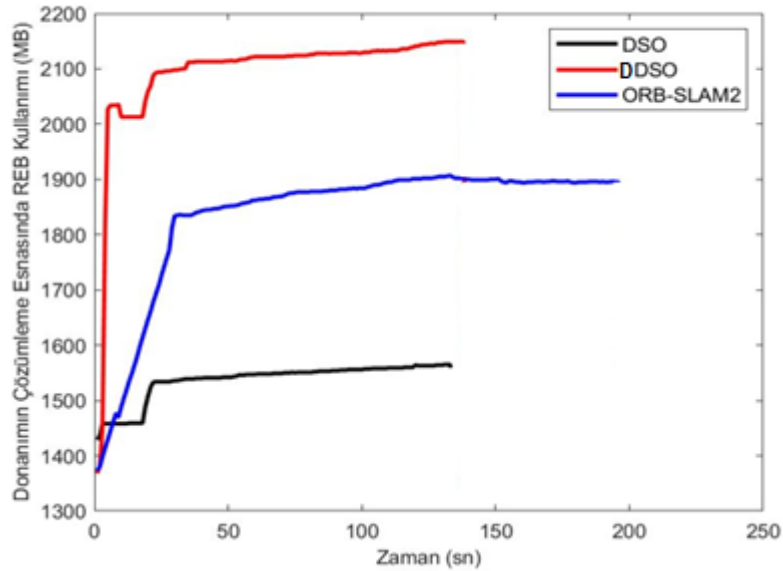
EZKH yöntemlerinin donanım kısıtlarına etkisinin değerlendirilmesi için REB yükünün de incelenmesi amaçlanmıştır. DSO, DDSO ve ORB-SLAM2 yöntemlerinin ICL-NUIM, KITTI, EUROK ve TUM TGO veri kümelerine ait video dizileri üzerinden oluşturduğu REB yükleri NVIDIA Jetson TX1 donanımının “tegrastats” yazılım arayüzü kullanılarak kaydedilmiştir.

ICL-NUIM veri kümesi üzerinde DSO, DDSO ve ORB-SLAM2 yöntemlerinin oluşturduğu REB yüklerinin kok değerleri Şekil 4.14 ile verilmektedir.

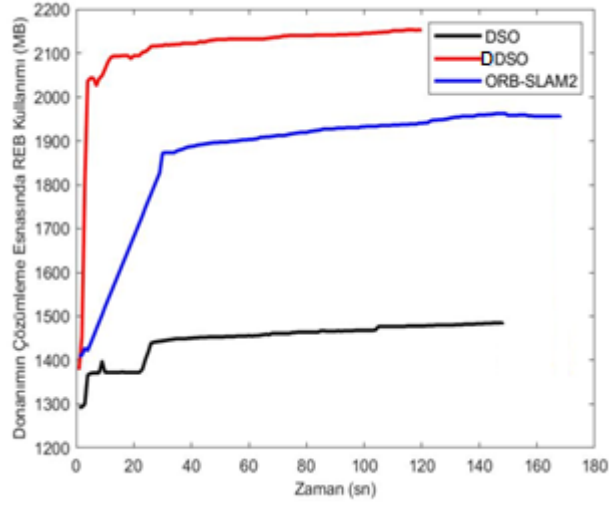


Şekil 4.14 ICL-NUIM veri kümesine ait officeRoom0 ve livingRoom0 video dizileri üzerinde EZKH yöntemlerinin REB Kullanımlarına etkileri (MB - kok)

Şekil 4.5 ve Şekil 4.6 ile ise EZKH yöntemlerinin çözümleme sürelerince donanım üzerinde oluşan REB yükü karşılaştırmaları verilmektedir. DSO'nun döngü kapama kabiliyetinin olmaması sebebiyle, REB yükünün diğer EZKH yöntemlerine kıyasla daha az olduğu görülmektedir.

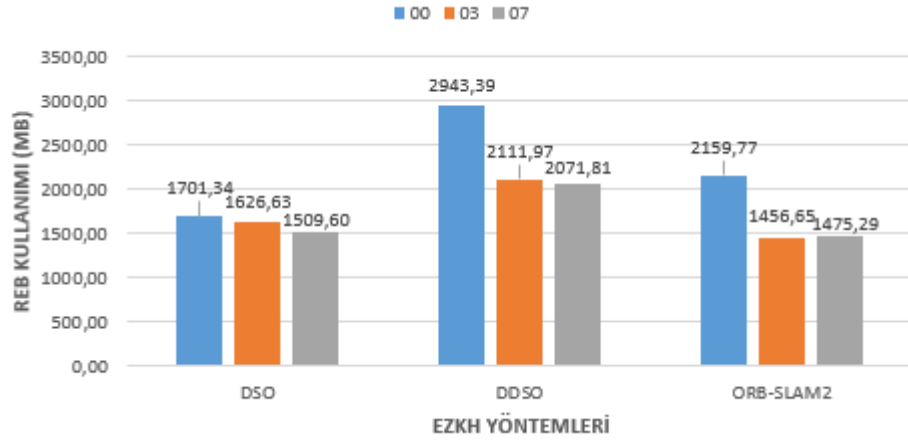


Şekil 4.15 ICL-NUIM veri kümesine ait livingRoom0 video dizisi - DSO, DDSO ve ORB-SLAM2 yöntemlerinin oluşturduğu REB yükleri



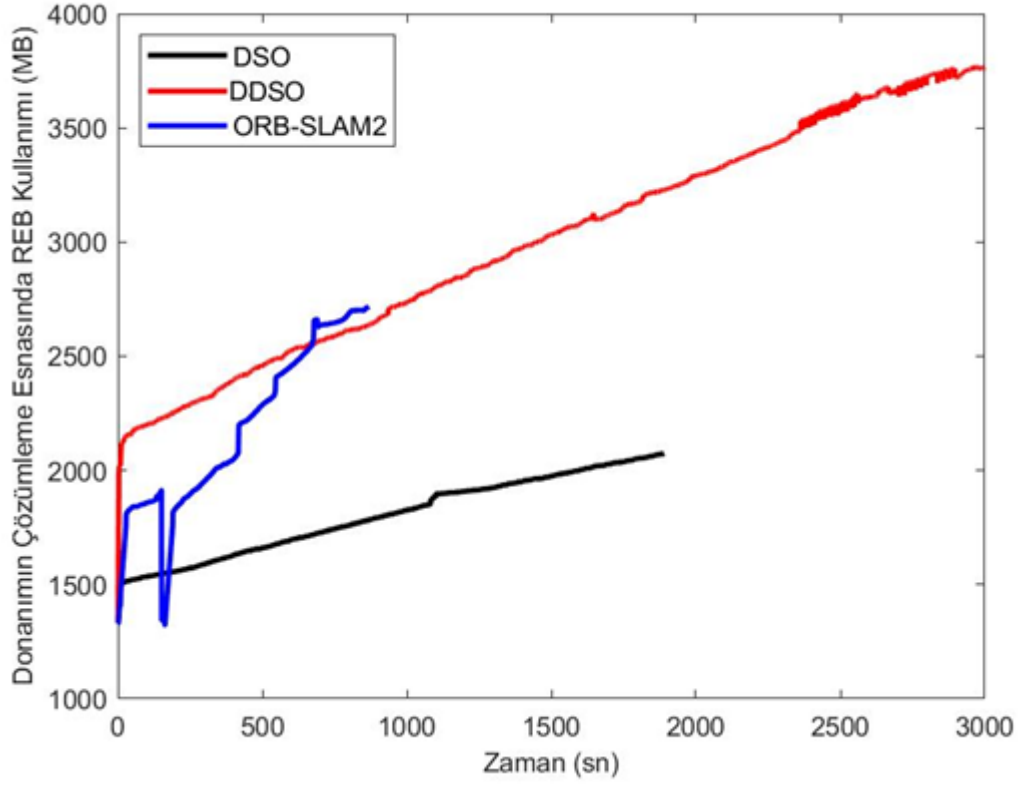
Şekil 4.16 ICL-NUIM veri kümesine ait officeRoom0 video dizisi - DSO, DDSO ve ORB-SLAM2 yöntemlerinin oluşturduğu REB yükleri

KITTI veri kümesi üzerinde DSO, DDSO ve ORB-SLAM2 yöntemlerinin REB kullanımı (kok) Şekil 4.17 ile verilmektedir. Tüm yöntemler için en fazla REB yükü, fazla sayıda video çerçevesine sahip olması sebebiyle 00 isimli video dizisinde görülmüştür.



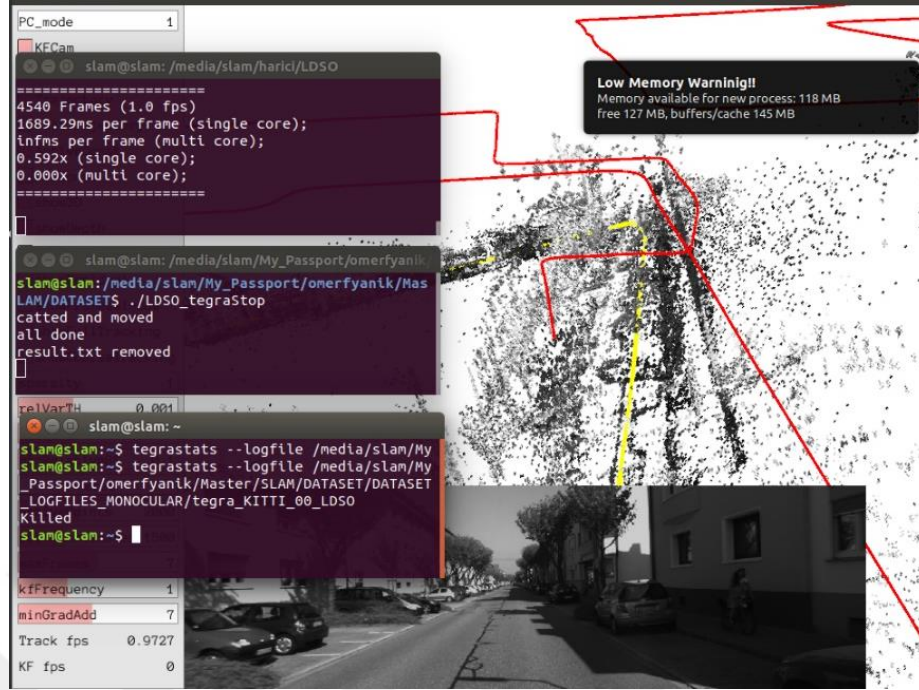
Şekil 4.17 DSO, DDSO ve ORB-SLAM2 yöntemlerinin KITTI veri kümesine ait 00, 03 ve 07 isimli video dizilerinde çözümü yaparken donanımda oluşan REB yükleri

Yöntemler arasında REB kullanımı en yüksek olan yöntem ise DDSO yöntemi olmuştur. Şekil 4.17 ve Şekil 4.18’de de görüleceği üzere 00 numaralı video dizisinde, DDSO yönteminin REB kullanımının diğer video dizilerindekine göre çok daha yüksek olmuştur.

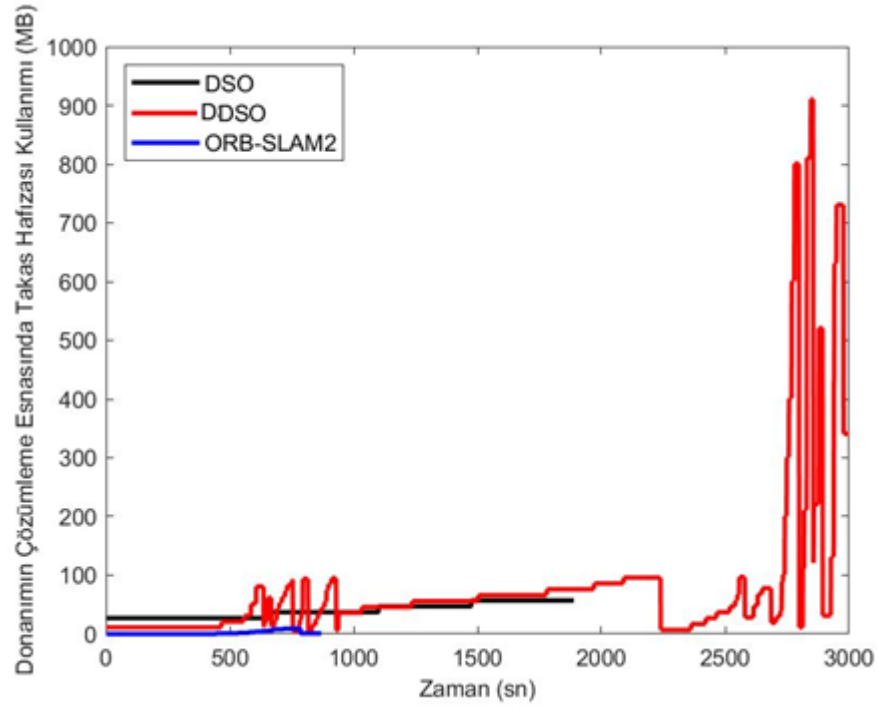


Şekil 4.18 DSO, DDSO ve ORB-SLAM2 yöntemlerinin KITTI 00 isimli video dizisi üzerinde çözümüleme yaparken donanımda oluşan REB yükü

Şekil 4.19’da görüleceği üzere KITTI 00 isimli video dizisi üzerinde çözümüleme yaparken fazla REB yükü oluşturmaları sebebiyle, DDSO yöntemi donanımın “Düşük Hafıza” uyarısı vermesine sebep olmuştur. Bu durum, REB kullanımına ek olarak, geçici hafıza kullanımını desteklemek üzere ayrılmış olan takas (swap) hafızası kullanımının DDSO yönteminde 00 video dizisi için diğer yöntemlerdekine kıyasla çok daha yüksek olmasını beraberinde getirmiştir (Şekil 4.20).

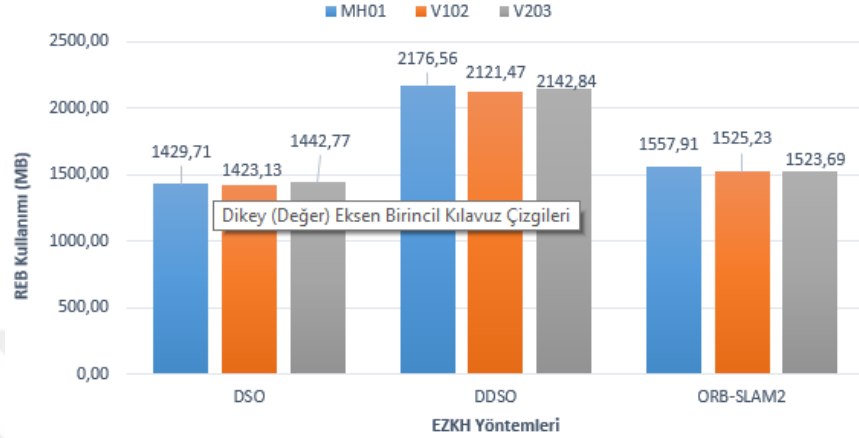


Şekil 4.19 DDSO Düşük Hafıza Uyarısı - KITTI 00



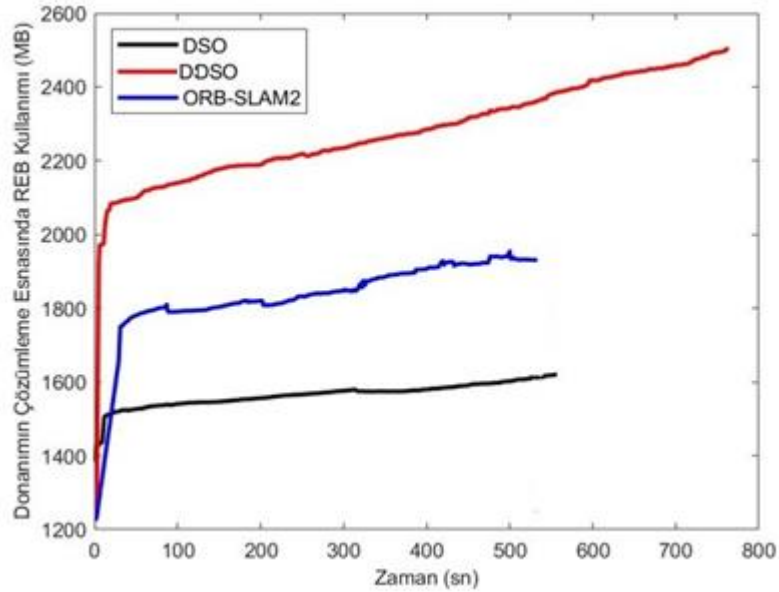
Şekil 4.20 KITTI 00 video dizisi üzerinde DSO, DDSO ve ORB-SLAM2 yönteminin oluşturduğu takas (swap) hafıza yükleri

EUROC veri kümesine ait MH01, V102 ve V203 isimli video dizileri üzerinde DSO, DDSO ve ORB-SLAM2 yöntemlerinin donanım üzerinde oluşturdukları REB yükleri Şekil 4.21 ile verilmektedir. DDSO yönteminin REB kullanımı açısından diğer yöntemlere göre daha zorlayıcı olduğu gözlenmiştir.



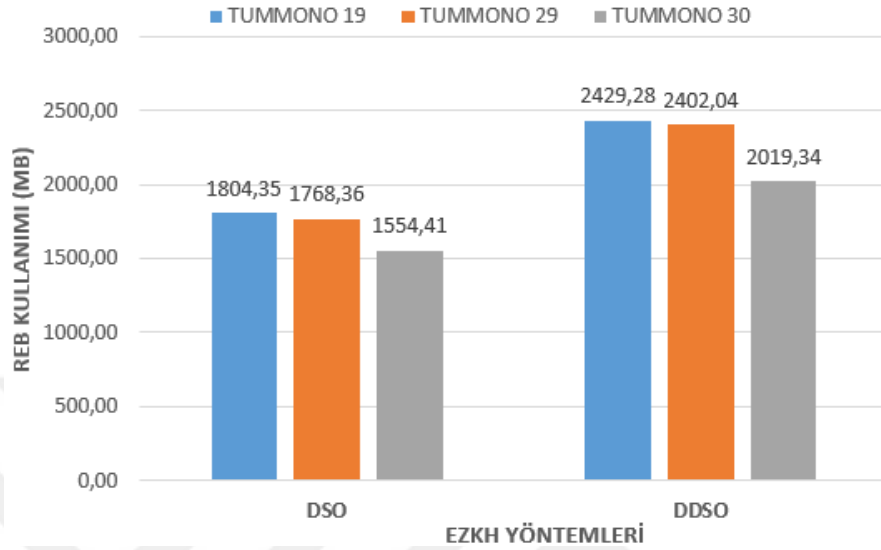
Şekil 4.21 EZKH yöntemlerinin EUROC veri kümesi üzerinde çözümleme yaparken donanım üzerinde oluşan REB yükü

Şekil 4.22’de MH01 veri kümesi üzerinde EZKH yöntemleri çözümleme yaparken donanımın zamana bağlı REB kullanımı karşılaştırmalı olarak verilmiştir.

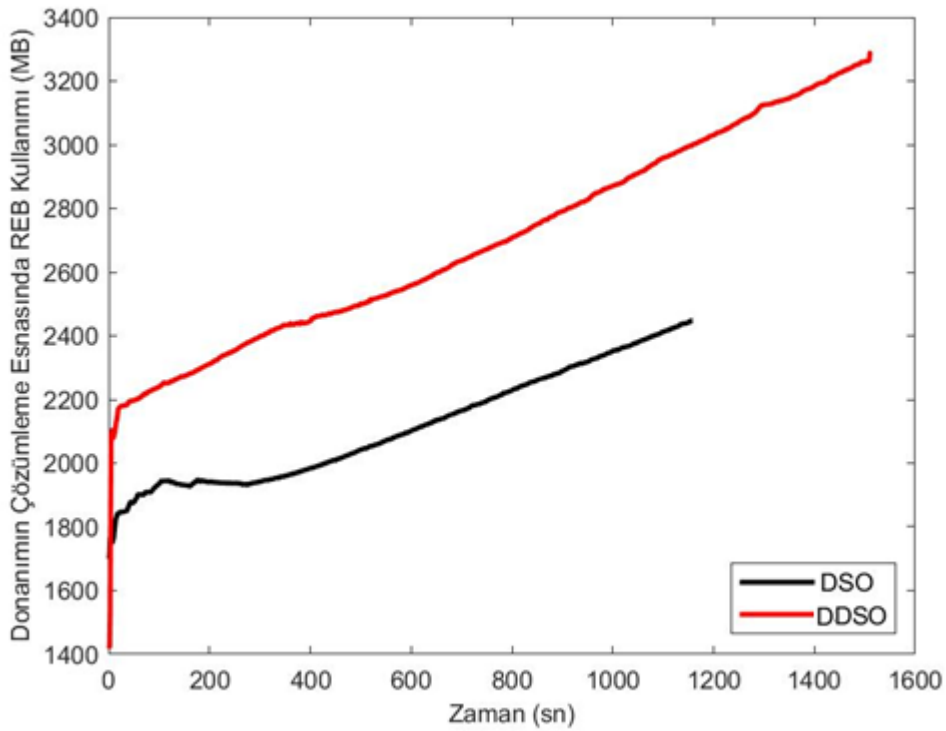


Şekil 4.22 EUROC veri kümesine ait MH01 video dizisi için çözümleme süresince REB kullanım davranışları

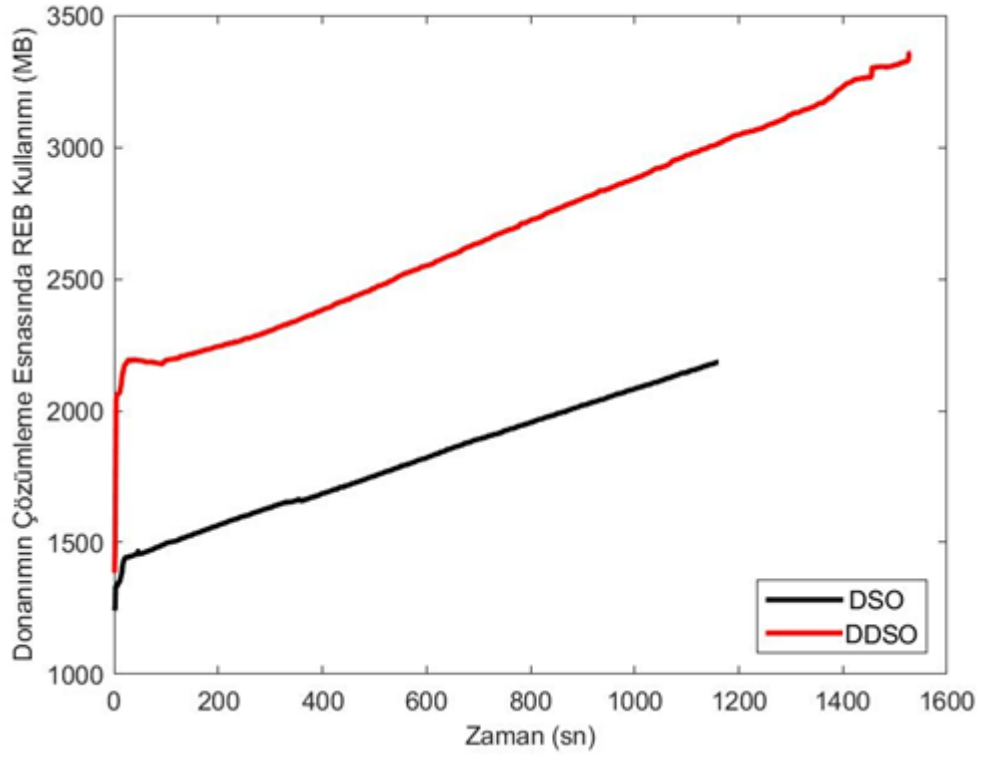
TUM TGO veri kümesi üzerinde DSO ve DDSO yöntemlerinin REB kullanımı Şekil 4.23, Şekil 4.24 ve Şekil 4.25 ve Şekil 4.26 ile verilmektedir. DDSO yönteminin REB kullanımının tüm veri kümelerinde daha zorlayıcı olduğu gözlenmiştir.



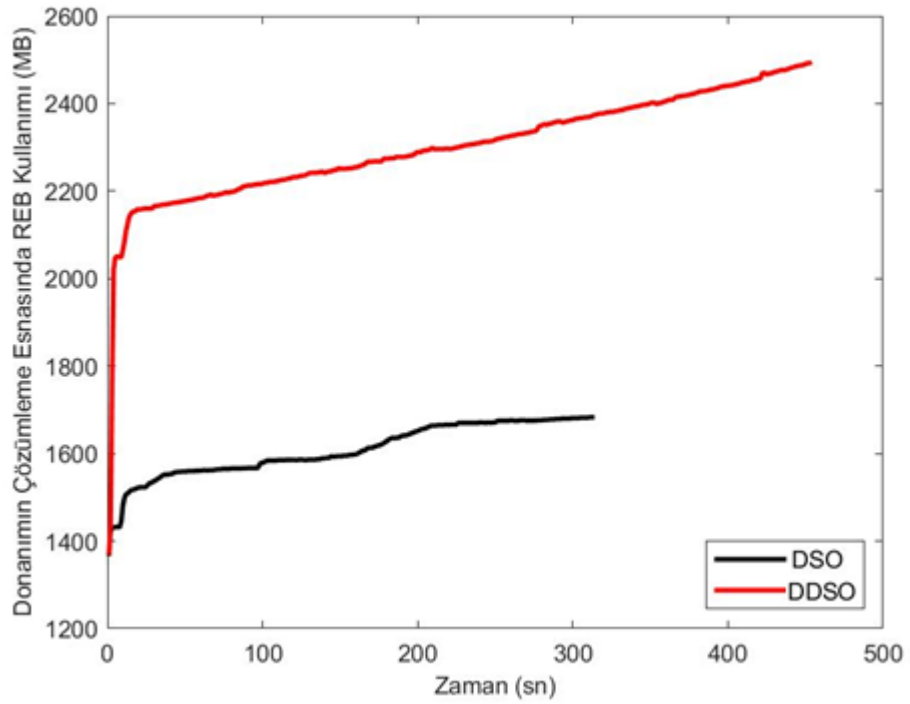
Şekil 4.23 TUM TGO veri kümesine ait video dizileri üzerinde DSO ve DDSO yöntemlerinin donanımın REB kullanımına etkileri



Şekil 4.24 TUM TGO 19 isimli video dizisi üzerinde zamana bağlı REB kullanımı



Şekil 4.25 TUM TGO 29 isimli video dizisi üzerinde zamana bağlı REB kullanımı

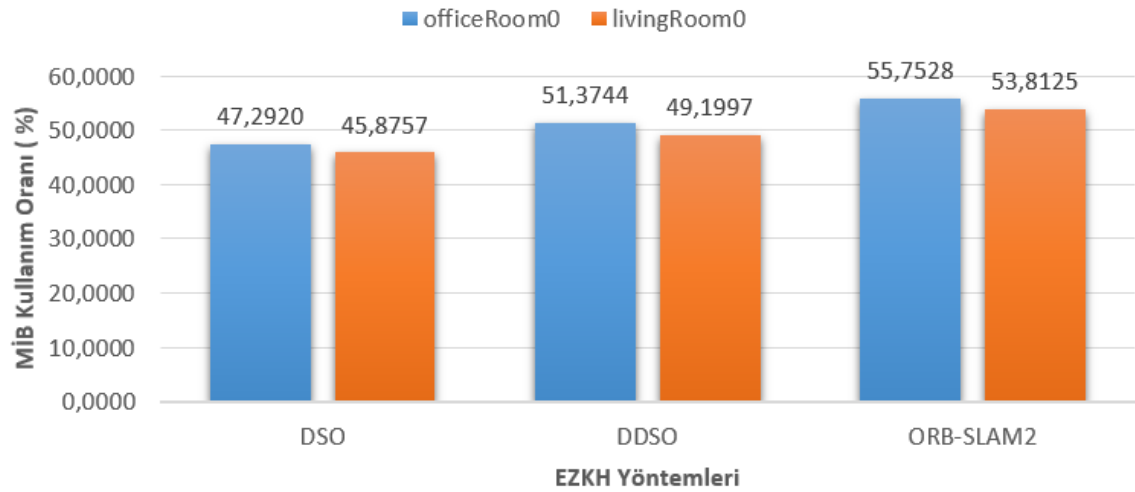


Şekil 4.26 TUM TGO 30 isimli video dizisi üzerinde zamana bağlı REB kullanımı

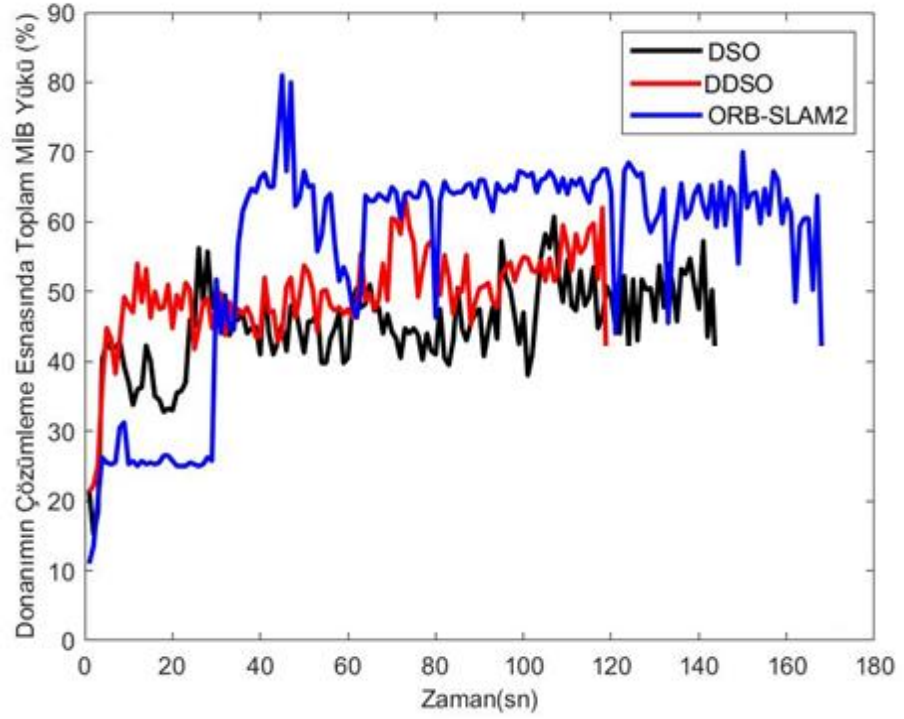
4.4 Merkezi İşlemci Birimi Kullanımı

EZKH yöntemlerinin işlem yükü hesaplamalarında sıklıkla kullanılan parametrelerden biri de algoritmanın donanım üzerinde oluşturduğu MİB yüküdür. Algoritmanın donanım üzerinde oluşturduğu MİB yükünün elde edilebilmesi için NVIDIA Jetson TX1 donanımının “tegrastats” yazılım arayüzü kullanılmıştır. TX1’in dört çekirdekli işlemciye sahip olması sebebiyle dört işlemcinin her birine ait kullanım oranlarının ortalaması alınarak karşılaştırmalar gerçekleştirilmiştir.

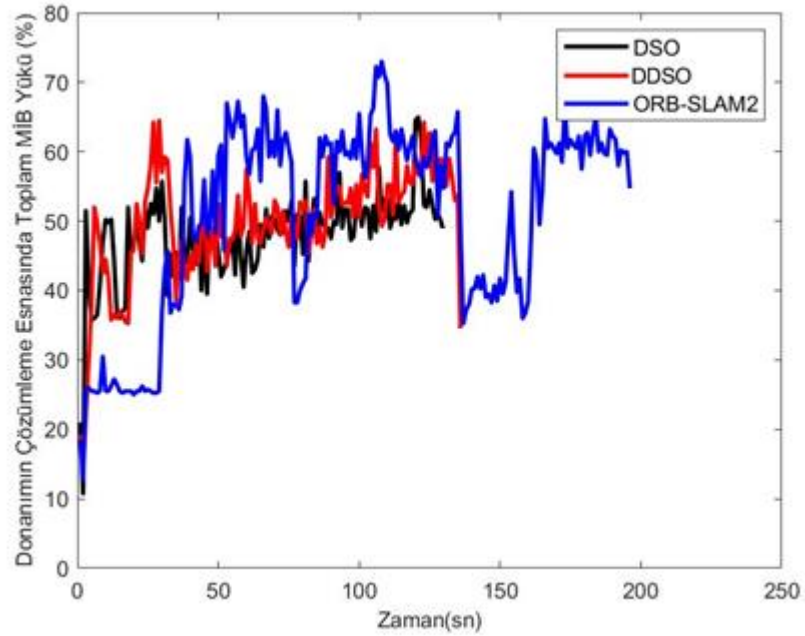
ICL-NUIM veri kümesine ait officeRoom0 ve livingRoom0 video dizileri üzerinde elde edilen sonuçlar Şekil 4.27, Şekil 4.28 ve Şekil 4.29 ile verilmektedir. DSO ve DDSO yöntemleri karşılaştırıldığında, REB kullanımında olduğu gibi MİB kullanımında da döngü kapama işleminin getirdiği fazla işlem yükü kendini göstermektedir.



Şekil 4.27 ICL-NUIM veri kümesi üzerinde EZKH yöntemlerinin çözümlemeleri esnasında donanımda oluşan MİB yükü (kok)

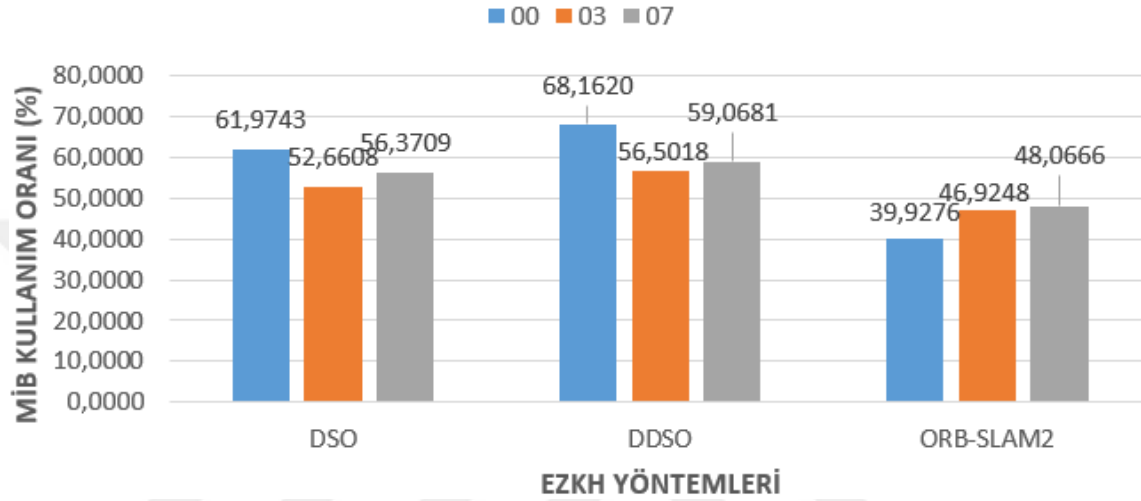


Şekil 4.28 ICL-NUIM officeRoom0 video dizisi üzerinde EZKH yöntemlerinin çözümlemeleri esnasında donanımda oluşan zamana bağlı MİB yükleri



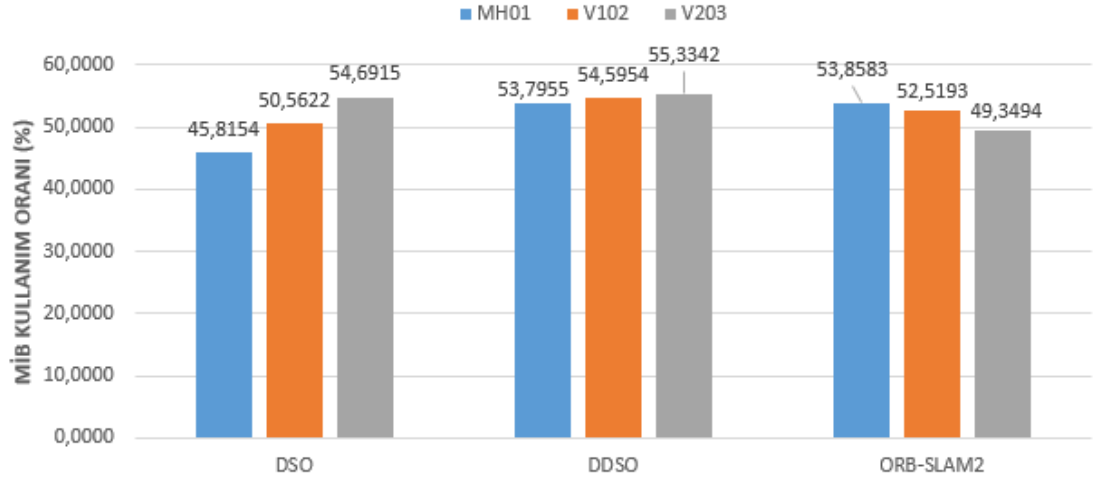
Şekil 4.29 ICL-NUIM livingRoom0 video dizisi üzerinde EZKH yöntemlerinin çözümlemeleri esnasında donanımda oluşan zamana bağlı MİB yükleri

KITTI veri kümesi üzerinde DSO, DDSO ve ORB-SLAM2 yöntemlerinin MİB kullanım yükleri Şekil 4.30 ile verilmektedir. DDSO ve DSO yöntemleri karşılaştırıldığında döngü kapama işleminin MİB kullanımına olan etkisi görülmektedir. Bununla birlikte KITTI veri kümesi özelinde değerlendirildiğinde ORB-SLAM2 yöntemi MİB kullanımı açısından diğer yöntemlere göre daha az maliyetli bir yöntem olarak ortaya çıkmaktadır.



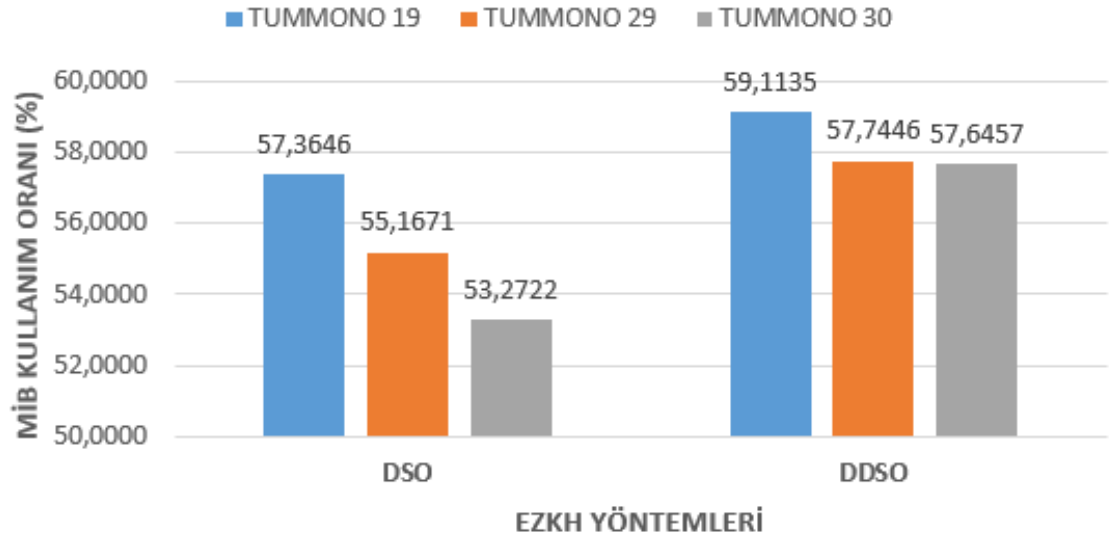
Şekil 4.30 KITTI veri kümesine ait 00, 03 ve 07 isimli video dizileri üzerinde EZKH yöntemlerinin çözümlemesi esnasında donanımda oluşan MİB yükleri (kok)

Şekil 4.31 ile EUROC veri kümesine ait video dizileri üzerinde DSO, DDSO ve ORB-SLAM2 yöntemlerinin donanımın MİB yüklerine etkileri (kok) verilmektedir. DDSO ve DSO yöntemleri karşılaştırıldığında döngü kapama işleminin MİB kullanımına olan etkisi EUROC veri kümesi üzerinde de görülmektedir. Bununla birlikte MH01 ve V102 video dizileri açısından en az MİB yükü getiren yöntem DSO iken, V203 video dizisi üzerinde ORB-SLAM2 en az MİB yükü getiren yöntem olmuştur. Ayrıca DDSO ve ORB-SLAM 2 yöntemlerinin MH01 ve V102’de birbirine yakın MİB kullanım oranları ortaya çıkmıştır.



Şekil 4.31 EUROC veri kümesine ait video dizileri üzerinde EZKH yöntemlerinin çözümlemesi esnasında donanımda oluşan MİB yükleri (kok)

DSO ve DDSO yöntemlerinin karşılaştırıldığı TUM TGO veri kümesi üzerinde DSO ve DDSO yöntemlerinin donanım üzerinde oluşturduğu MİB yükleri (kok) Şekil 4.32 ile verilmektedir. Tüm video dizileri üzerinde DDSO yöntemi MİB üzerinde daha fazla yük oluşturmuştur. Bu haliyle, DDSO ve DSO yöntemleri karşılaştırıldığında döngü kapama işleminin MİB kullanımına olan etkisi görülmektedir.



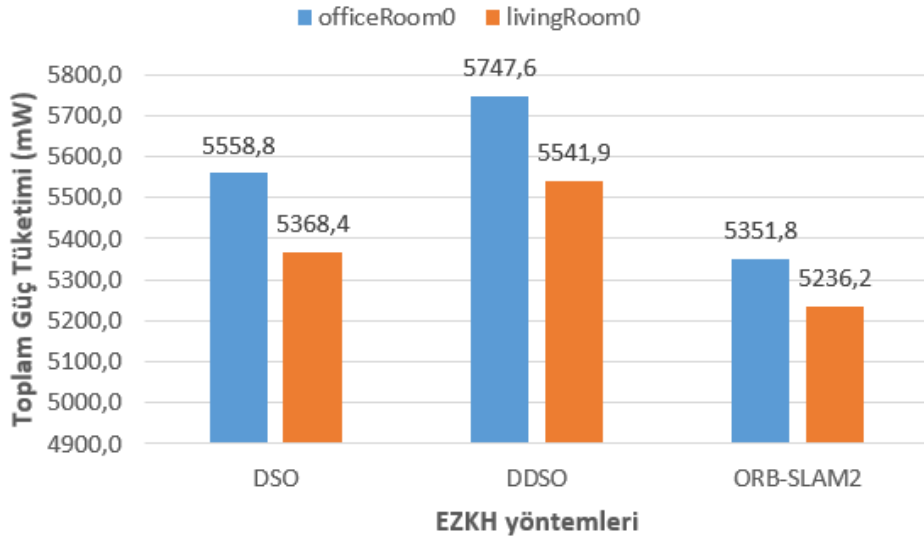
Şekil 4.32 TUM TGO veri kümesine ait video dizileri üzerinde EZKH yöntemlerinin getirdiği MİB yükleri (kok)

4.5 Toplam Güç Tüketimi

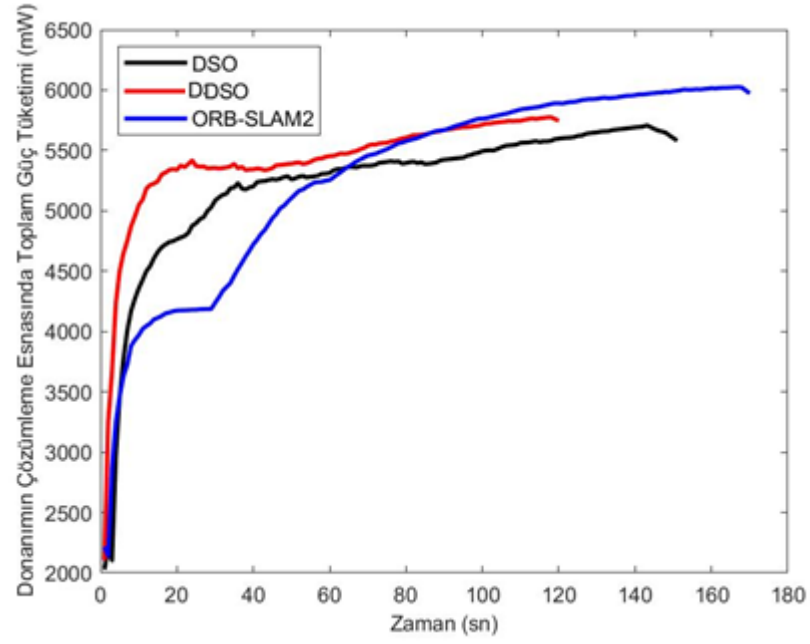
Otonom sistemler açısından en önemli kısıtlardan bir tanesi sınırlı güç kaynaklarıdır. Bu sebeple MİB üzerinde oluşan işlem yüküne ek olarak donanımın toplam güç tüketiminin EZKH yöntemlerine bağlı olarak nasıl değiştiğinin de incelenmesi doğru bir yaklaşımdır olacaktır.

NVIDIA Jetson TX1 donanımı MİB gücü, Grafik İşlemci Birimi (GİB) gücü ve toplam güç harcamalarını “tegrastats” yazılım arayüzünden vermektedir. EZKH yöntemlerinin donanımın güç ihtiyaçlarını nasıl etkileyeceği değerlendirilmek istendiğinden tegrastats arayüzünden sağlanan toplam güç tüketimi değeri dikkate alınmıştır.

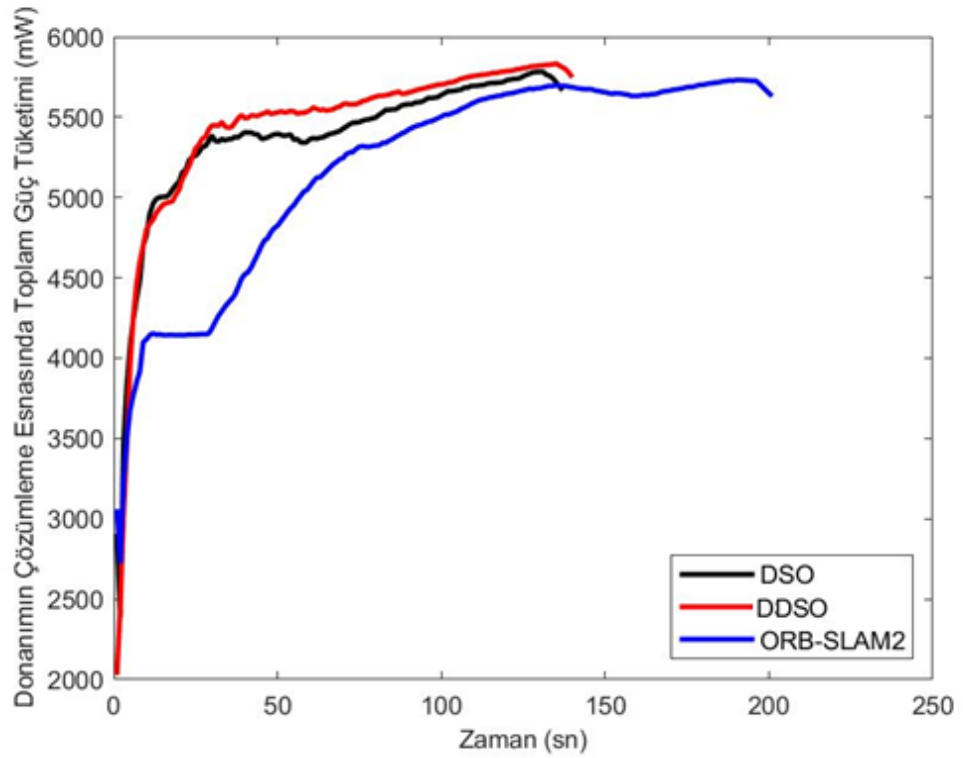
ICL-NUIM veri kümesine ait officeRoom0 ve livingRoom0 video dizileri kullanıldığında DSO, DDSO ve ORB-SLAM2 yöntemlerinin NVIDIA Jetson TX1 donanımı üzerindeki toplam güç tüketimine etkileri (kok) Şekil 4.33 ile verilmiştir. Ayrıca video dizilerinin çözülmesi esnasında zamana bağlı olarak donanımın tükettiği güçler Şekil 4.34 ve Şekil 4.35 ile verilmektedir. Görüldüğü üzere livingRoom0 ve officeRoom0 video dizileri üzerinde en az güç tüketimi ORB-SLAM2 ile sağlanmışken, bu haliyle EZKH yöntemlerinin toplam güç tüketimlerine olan etkileri, MİB kullanımına olan etkiden (Şekil 4.28) farklılık göstermiştir.



Şekil 4.33 ICL-NUIM veri kümesi üzerinde DSO, DDSO ve ORB-SLAM2 yöntemlerinin güç tüketimine etkileri (kok)

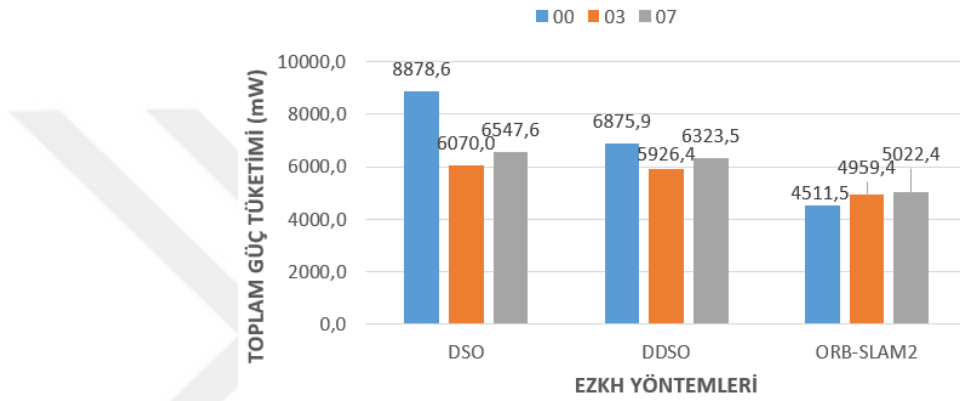


Şekil 4.34 ICL-NUIM officeRoom0 video dizisi üzerinde DSO, DDSO ve ORB-SLAM2 yöntemlerinin zamana bağlı güç tüketimine etkisi

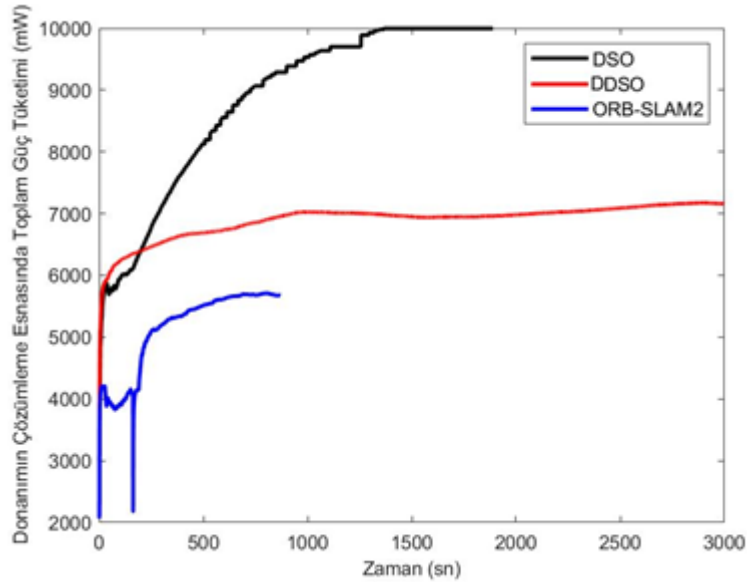


Şekil 4.35 ICL-NUIM livingRoom0 video dizisi üzerinde DSO, DDSO ve ORB-SLAM2 yöntemlerinin zamana bağlı güç tüketimine etkisi

KITTI veri kümesine ait 00, 03 ve 07 numaralı video dizileri üzerinde DSO, DDSO ve ORB-SLAM2 yöntemlerinin sebep olduğu güç tüketimlerine ilişkin karşılaştırmalar Şekil 4.36 ile verilmektedir. Tüm veri kümeleri açısından en etkin yöntem ORB-SLAM2 iken, en fazla güç harcanmasına sebep olan yöntem ise DSO yöntemi olmuştur. Bununla birlikte MİB yükleri (Şekil 4.30) ile kıyaslandığında, 00 veri kümesinde DSO yönteminin sebep olduğu güç tüketiminin, DDSO ve ORB-SLAM2'ye göre çok daha fazla olduğu dikkat çekmektedir. Şekil 4.37 ile EZKH yöntemlerinin 00 video dizisi üzerinde zamana bağlı olarak toplam güç tüketimine etkileri verilmektedir.

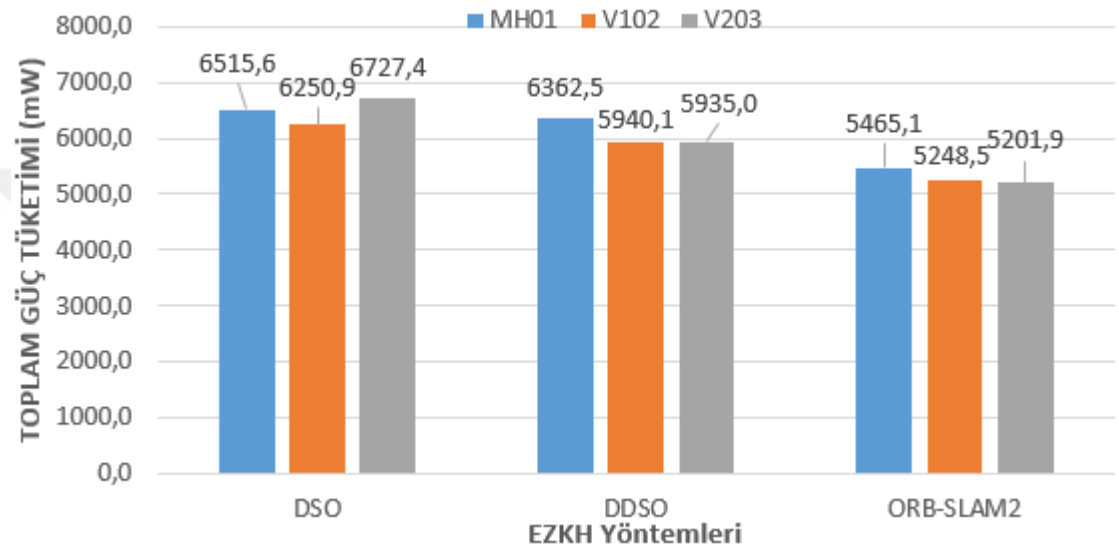


Şekil 4.36 KITTI veri kümesine ait video dizileri üzerinde EZKH yöntemlerinin güç tüketimine etkisi (kok)



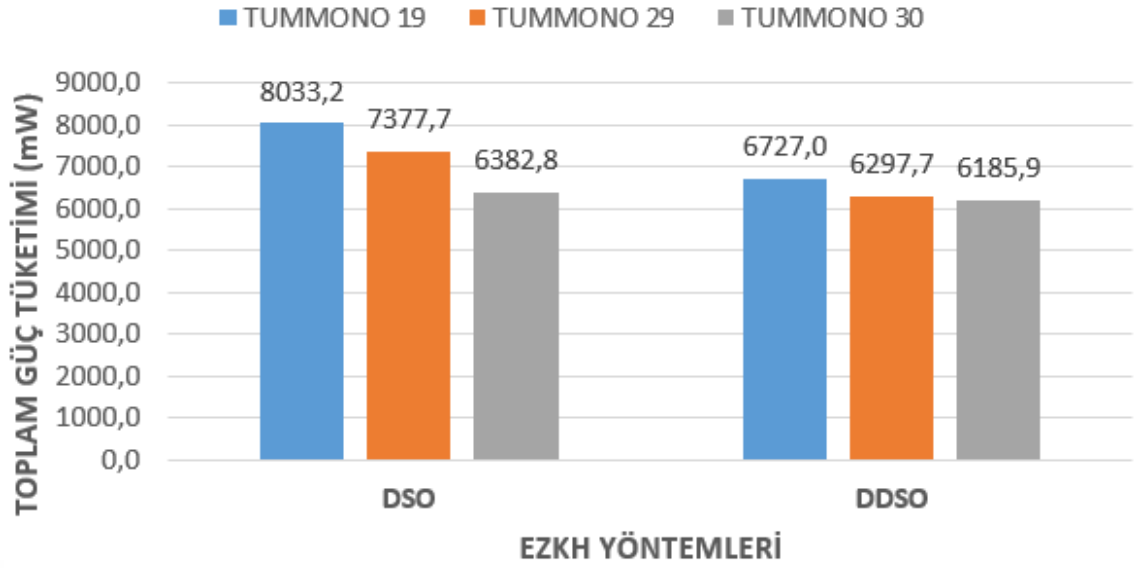
Şekil 4.37 KITTI 00 video dizisinin çözülmesi esnasında donanımın zamana bağlı güç tüketimi

EUROC veri kümesine ait MH01, V102 ve V203 isimli video dizileri üzerinde DSO, DDSO ve ORB-SLAM2 yöntemlerinin oluşturduğu güç tüketimine ilişkin karşılaştırmalar Şekil 4.38 ile verilmektedir. Tüm veri kümeleri açısından en az güç tüketimine sebep olan yöntem ORB-SLAM2 iken, en fazla güç harcanmasına sebep olan yöntem ise DSO yöntemi olmuştur. Ayrıca bu haliyle EUROC veri kümesi üzerinde güç tüketim davranışları MİB yükleri (Şekil 4.31) ile uyumluluk göstermemiştir.



Şekil 4.38 EUROC veri kümesine ait video dizileri üzerinde DSO, DDSO ve ORB-SLAM2 yöntemlerinin donanımın güç tüketimine etkileri (mW - kok)

TUM TGO veri kümesi üzerinde DSO ve DDSO yöntemleri karşılaştırılmış ve donanımın güç tüketimlerine olan etkileri incelenmiştir. TUM TGO veri kümesine ait video dizileri üzerinde toplam güç tüketimine ilişkin karşılaştırmalar Şekil 4.39 ile verilmektedir. Şekilden de görüleceği üzere tüm video dizilerinde DSO yöntemi DDSO yöntemine göre daha fazla güç tüketimine sebep olmaktadır. Diğer veri kümelerine ait video dizilerinde olduğu gibi, TUM TGO veri kümesine ait video dizilerinde de toplam güç tüketimi değerleri MİB yükleri (Şekil 4.32) ile paralellik göstermemektedir.

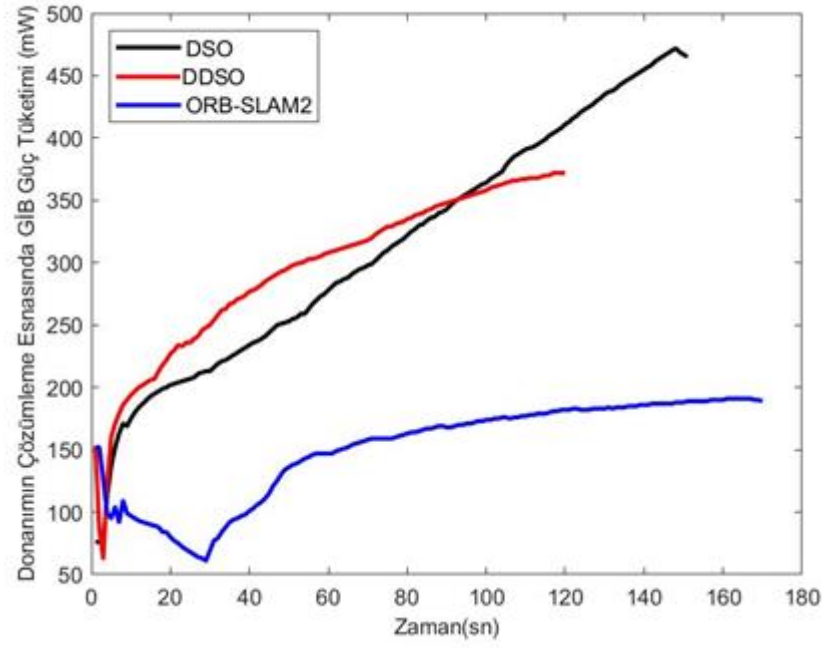


Şekil 4.39 TUM TGO veri kümesine ait video dizileri üzerinde DSO, DDSO ve ORB-SLAM2 yöntemlerinin donanımın güç tüketimine etkileri (mW - rms)

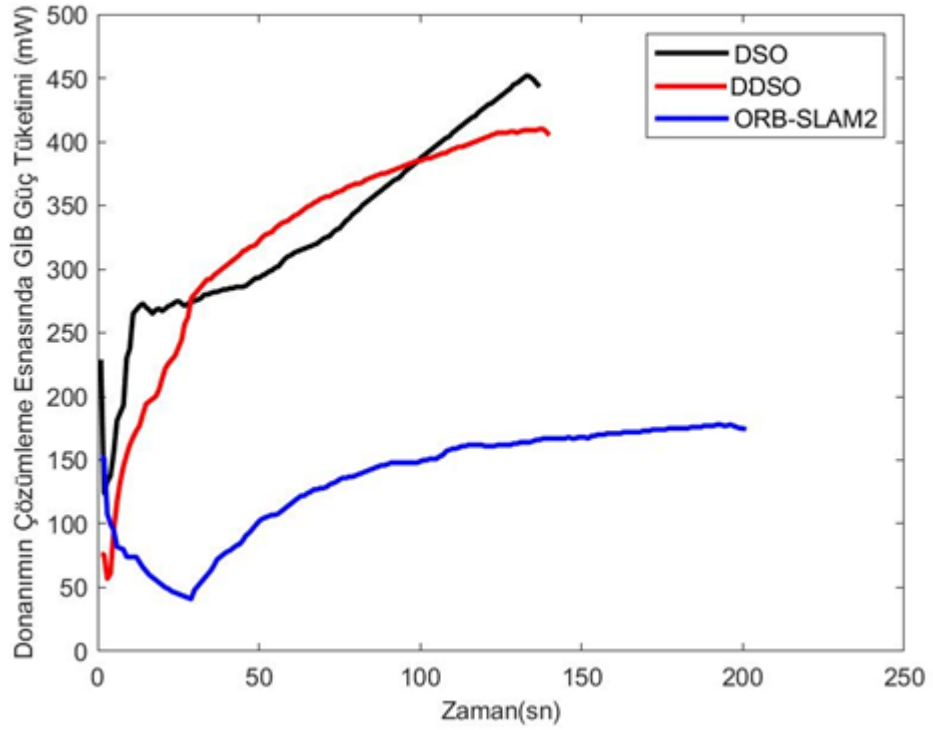
4.6 Grafik İşlemci Birimi Güç Tüketimi

EZKH yöntemlerinin video dizileri üzerinde oluşturduğu MİB yükleri ile toplam güç tüketimi davranışlarının farklı olması, GİB güç tüketimlerini de kontrol etme ihtiyacını ortaya çıkarmıştır. NVIDIA Jetson TX1 donanımının sağladığı veriler ışığında elde edilen sonuçlar toplam güç tüketiminde MİB yükleri ile olan çelişkiyi açıklar nitelikte olmuştur.

ICL-NUIM veri kümesi üzerinde GİB'in harcadığı güç miktarı analiz edilmiştir. officeRoom0 ve livingRoom0 video dizisi üzerinde DSO ve DDSO yöntemlerinin ORB-SLAM2 yöntemine kıyasla daha fazla GİB kullanımı olduğu gözlenmiştir. ICL-NUIM veri kümesi üzerinde GİB güç tüketimine ilişkin karşılaştırma grafikleri Şekil 4.40 ve Şekil 4.41 ile verilmektedir. Elde edilen sonuçlar MİB yükü (Şekil 4.27)ve toplam güç tüketimleri (Şekil 4.33) ile kıyaslandığında oluşan farklılığın sebebinin DSO yönteminin GİB bakımından çok daha fazla yük oluşturmasıdır.

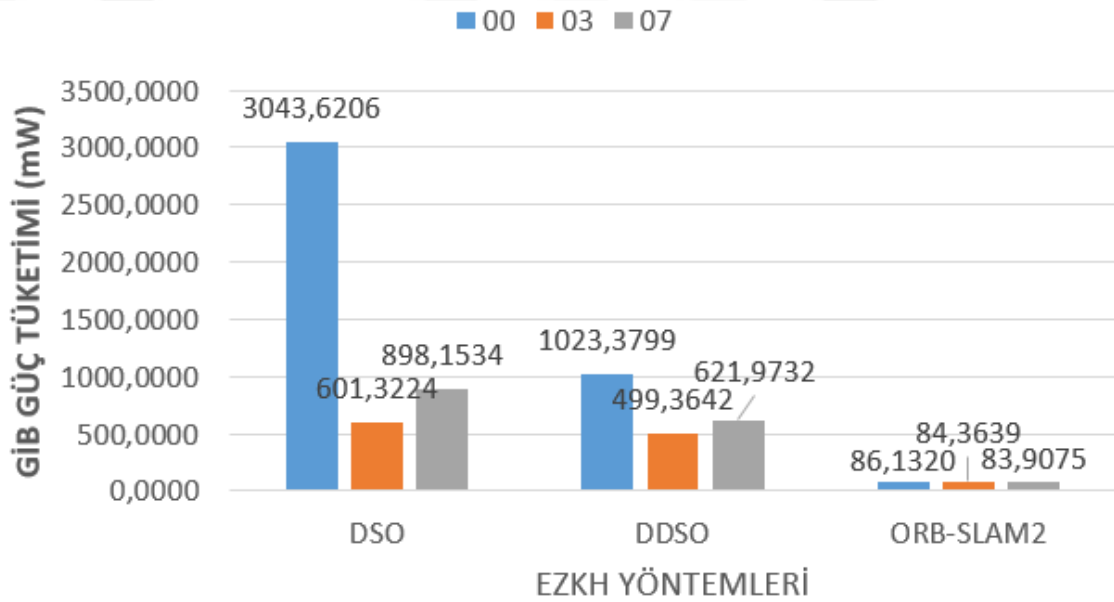


Şekil 4.40 ICL-NUIM veri kümesine ait officeRoom0 video dizisi üzerinde zamana GİB güç tüketim karşılaştırmaları

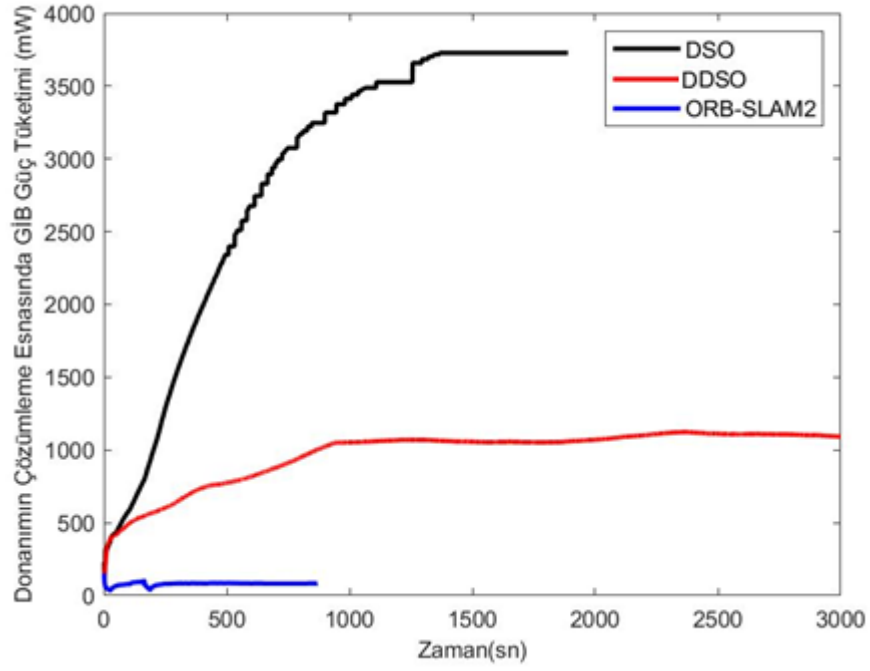


Şekil 4.41 ICL-NUIM veri kümesine ait livingRoom0 video dizisi üzerinde zamana GİB güç tüketim karşılaştırmaları

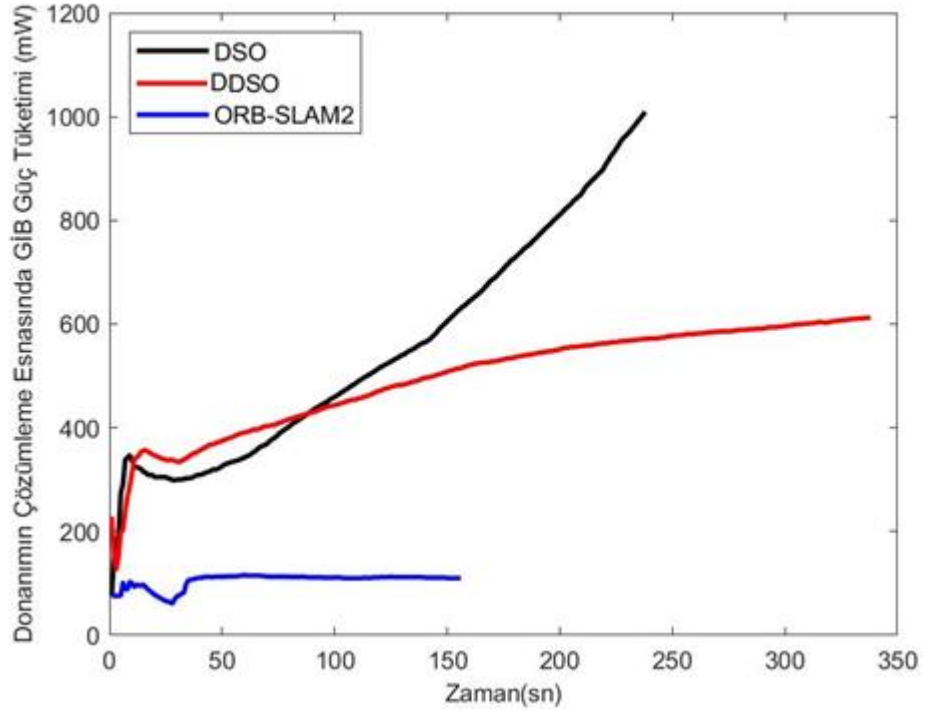
KITTI veri kümesine ait 00, 03 ve 07 isimli video dizilerinde zamana bağlı GİB kullanımı açısından yöntemler arasında yapılan karşılaştırma Şekil 4.42’de verilmiştir. Her üç veri kümesi için bakıldığında GİB kullanımı açısından en maliyetli yöntemin DSO olduğu görülmektedir. Bununla birlikte video çerçevelerinin (kayıt süresinin) az olduğu 03 numaralı veri kümesinde DSO yönteminin harcadığı GİB gücünün DDSO ile kıyaslanabilir bir seviyede olduğu görülmüştür. Sürenin uzaması durumunda DSO’nun harcayacağı GİB gücünün DDSO ve ORB-SLAM2’nin harcayacağı GİB gücünden ıraksama eğiliminde olduğu görülmektedir (Şekil 4.43, Şekil 4.44, Şekil 4.45). ORB-SLAM2 yöntemi ise KITTI veri kümesine ait tüm video dizileri için GİB kullanımı açısından maliyet-etkin bir yöntem olarak kendini göstermektedir.



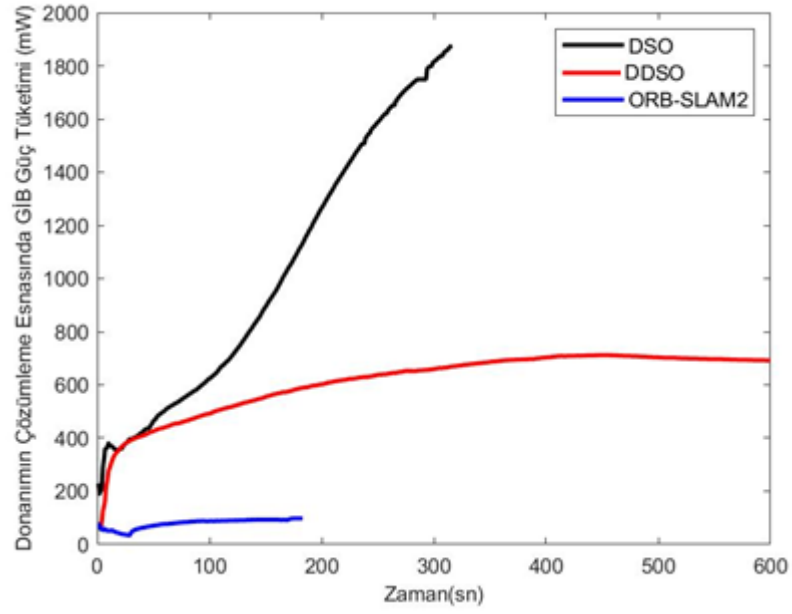
Şekil 4.42 KITTI veri kümesine ait video dizileri üzerinde çözümleme yapan EZKH yöntemlerinin GİB Güç Tüketimine etkileri (kok)



Şekil 4.43 KITTI veri kümesine ait 00 isimli video dizileri üzerinde EZKH yöntemlerinin GİB güç tüketimine zamana bağlı etkisi tüketimi

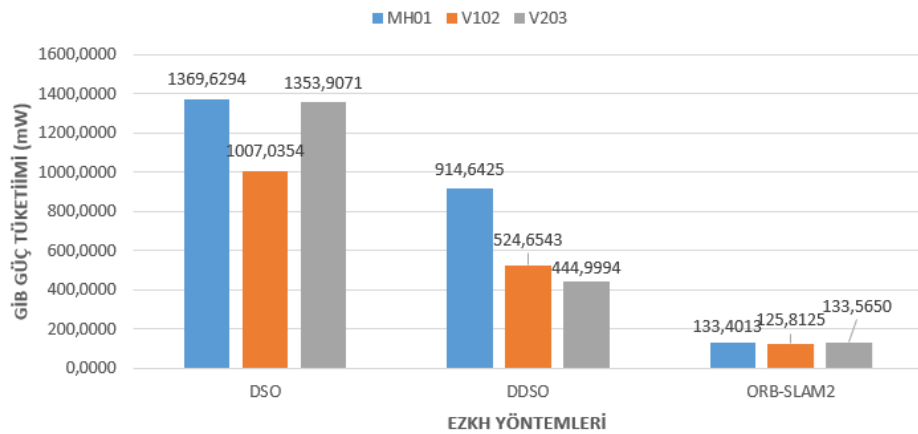


Şekil 4.44 KITTI veri kümesine ait 03 isimli video dizileri üzerinde EZKH yöntemlerinin GİB güç tüketimine zamana bağlı etkisi tüketimi

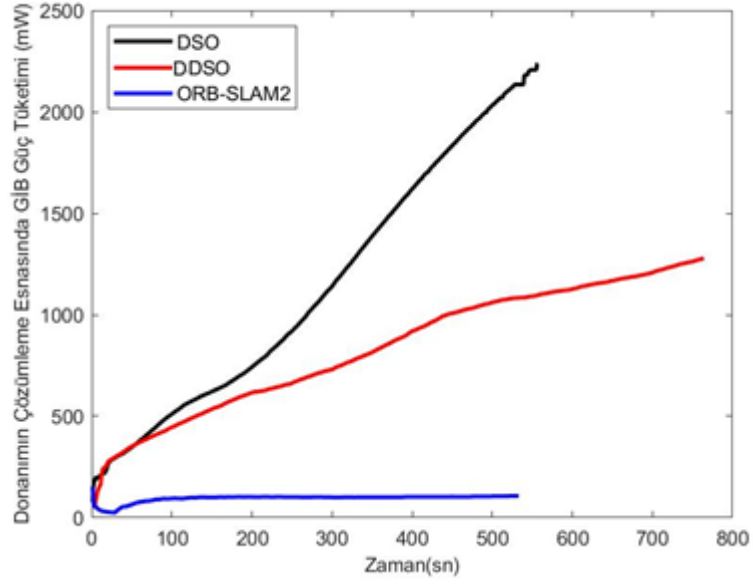


Şekil 4.45 KITTI veri kümesine ait 07 isimli video dizileri üzerinde EZKH yöntemlerinin GİB güç tüketimine zamana bağlı etkisi

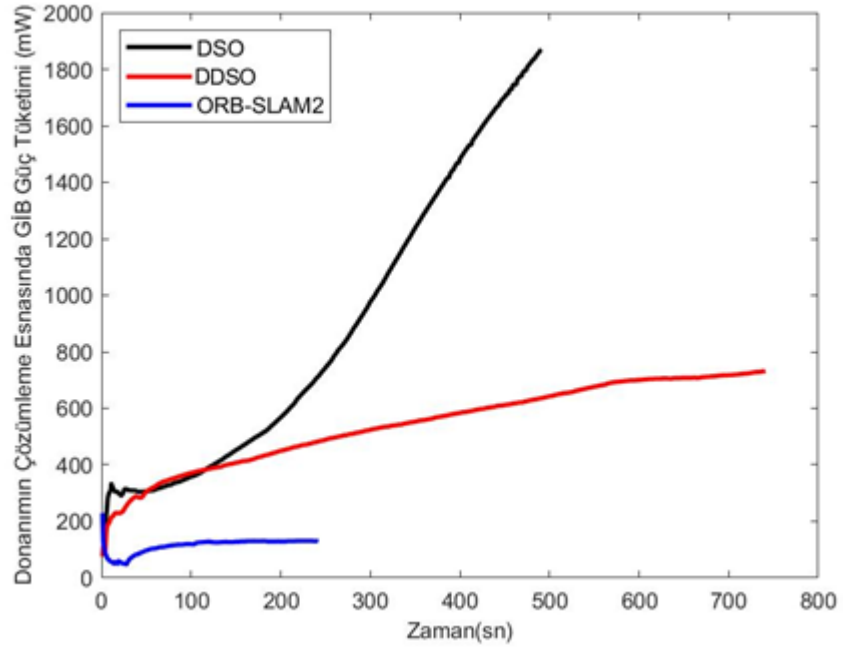
EUROC veri kümesine ait MH01, V102 ve V203 isimli video dizileri üzerinde EZKH yöntemlerinin GİB güç tüketimi açısından karşılaştırılmalarına ait grafikler Şekil 4.46, Şekil 4.47, Şekil 4.48 ve Şekil 4.49’de verilmiştir. Her üç veri kümesi için bakıldığında GİB güç tüketimi açısından en maliyetli yöntemin DSO olduğu görülmektedir. ORB-SLAM2 yöntemi ise EUROC veri kümesine ait video dizileri için GİB güç tüketimi açısından maliyet-etkin bir yöntem olarak kendini göstermektedir.



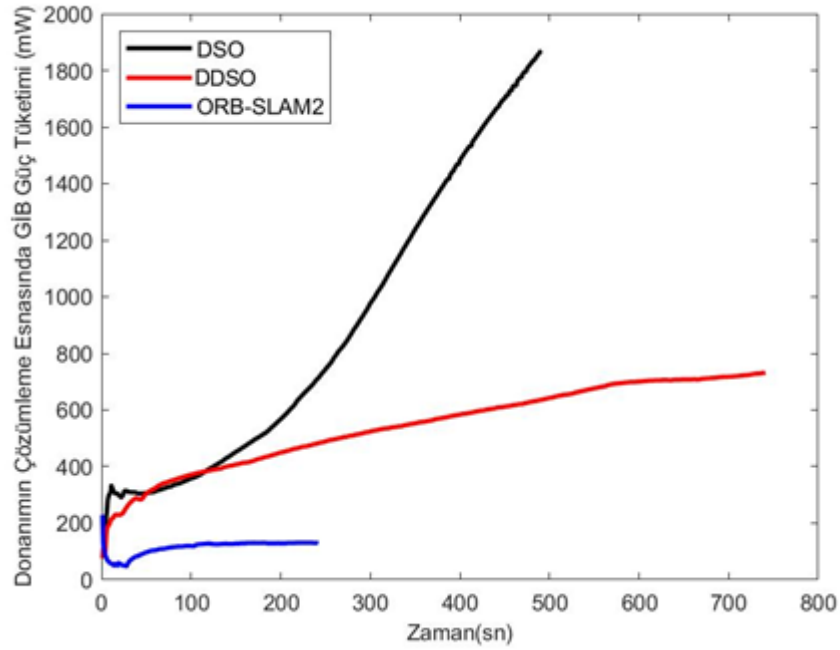
Şekil 4.46 EUROC veri kümesine ait video dizileri üzerinde EZKH yöntemlerinin donanımın GİB güç tüketimine etkisi (mW - kok)



Şekil 4.47 EUROC veri kümesine ait MH01 isimli video dizisinin çözülmesi esnasında donanımın GİB Güç Tüketimi

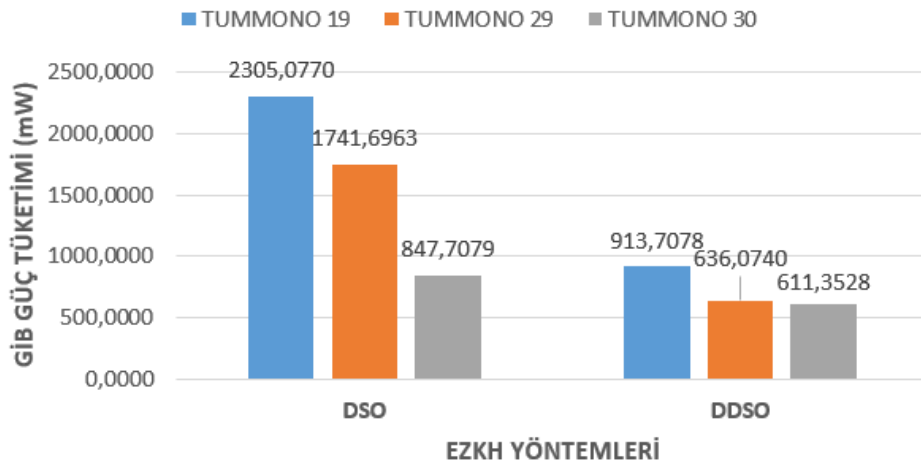


Şekil 4.48 EUROC veri kümesine ait V102 isimli video dizisinin çözülmesi esnasında donanımın GİB Güç Tüketimi (mW)

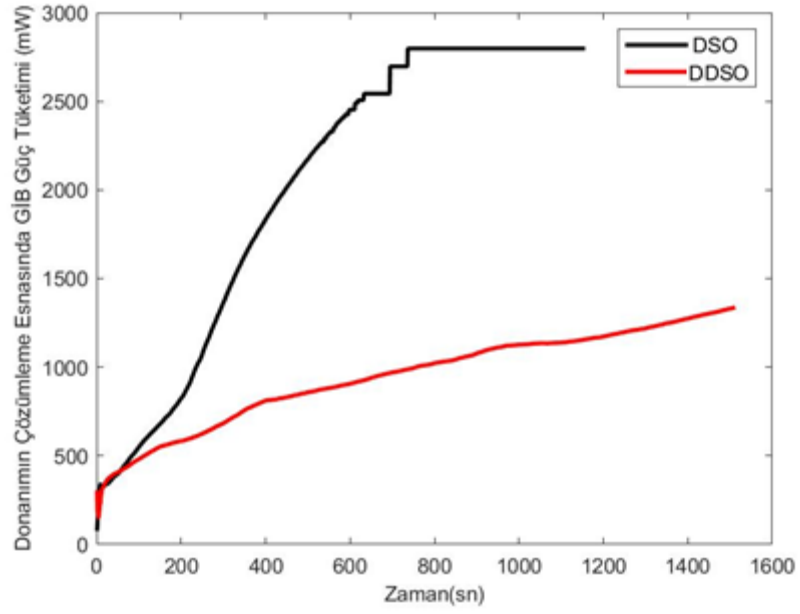


Şekil 4.49 EUROC veri kümesine ait V203 isimli video dizisinin çözülmesi esnasında donanımın GİB Güç Tüketimi (mW)

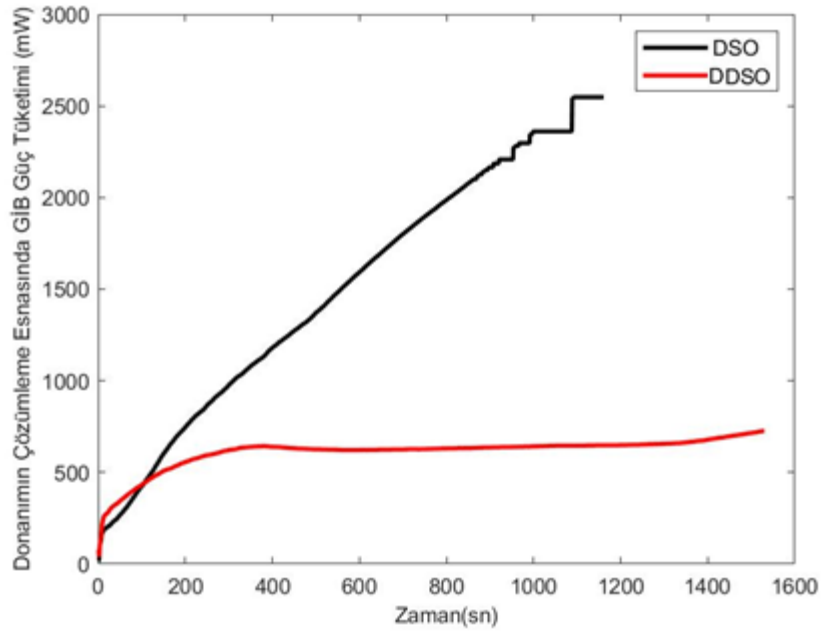
TUM TGO veri kümesine ait 19, 29 ve 30 isimli video dizileri üzerinde DSO ve DDSO yöntemlerinin GİB güç tüketimi açısından karşılaştırılmalarına ilişkin Şekil 4.50, Şekil 4.51, Şekil 4.52 ve Şekil 4.53 incelendiğinde GİB güç tüketimi açısından DSO yönteminin DDSO yöntemine göre daha maliyetli olduğu görülmektedir.



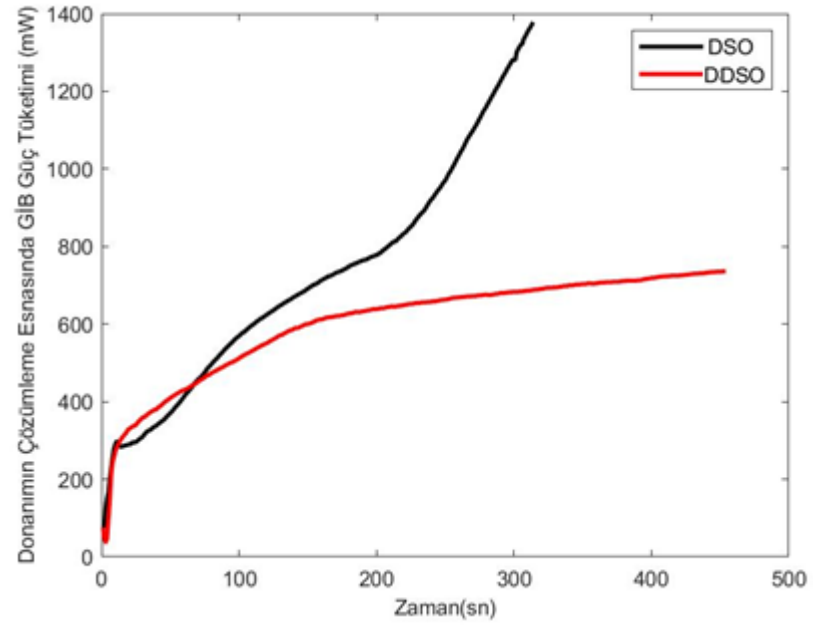
Şekil 4.50 TUM TGO veri kümesi üzerinde DSO ve DDSO yöntemlerinin çözülmesi esnasında donanımın GİB güç tüketimi (mW - kok)



Şekil 4.51 TUM TGO veri kümesi 19 isimli video dizisi üzerinde DSO ve DDSO yöntemlerinin çözümlemesi esnasında donanımın zamana bağlı GİB güç tüketimi



Şekil 4.52 TUM TGO veri kümesi 29 isimli video dizisi üzerinde DSO ve DDSO yöntemlerinin çözümlemesi esnasında donanımın zamana bağlı GİB güç tüketimi (mW)



Şekil 4.53 TUM TGO veri kümesi 30 isimli video dizisi üzerinde DSO ve DDSO yöntemlerinin çözümü esnasında donanımın zamana bağlı GİB güç tüketimi (mW)

5. TARTIŞMA VE SONUÇ

Bu tez çalışması kapsamında ARM mimaride bir merkezi işlemci bilgisayarı üzerinde çalışmaya uygun öznitelik tabanlı EZKH yöntemlerinden ORB-SLAM2 ile doğrudan yöntemlerden olan DSO ve DDSO yöntemlerinin performansları açık kaynaklarda yer alan ICL-NUIM, KITTI, EUROC ve TUM TGO veri kümeleri kullanılarak karşılaştırılmıştır. Bu haliyle öznitelik tabanlı yöntemler ile doğrudan yöntemlerin kıyaslaması yapılabilmişken, döngü kapama yeteneğinin getirdiği maliyetlerin de ortaya konulması hedeflenmiştir. Özellikle mobil robotların güç kısıtları ve kapalı mekânlarda konumlama ihtiyaçlarının karşılanabilmesi açısından yöntemler arasında tercih yapılabilmesi ve sonraki çalışmalara ışık tutulması amaçlanmıştır.

Karşılaştırma çalışmaları NVIDIA Jetson TX1 donanımı üzerinde aynı şartlar altında 6 kez gerçekleştirilmiştir. ORB-SLAM2, DSO ve DDSO yöntemleri,

- ICL-NUIM veri kümesi ile sağlanan en fazla video çerçevesine sahip olan officeRoom0 ve livingroom0 isimli video dizileri üzerinde,
- EUROC veri kümesi ile sağlanan ve her biri farklı zorluk tanımlamalarına (kolay, orta, zor) karşılık gelen MH01_Easy, V102_Medium ve V203_Difficult isimli video dizileri üzerinde,
- KITTI veri kümesi ile sağlanan her biri farklı kalibrasyon parametrelerine ve video çerçevesi sayısına sahip olan 00, 03, 07 isimli video dizileri üzerinde

karşılaştırılmıştır.

Ek olarak balıkgözü kamera verilerine sahip olan TUM TGO veri kümesi üzerinde DSO ve DDSO yöntemleri karşılaştırılmıştır.

Gerçekleştirilen karşılaştırma çalışmalarında Mutlak Yörünge Hatası, video çerçevesi başına çözümleme süresi, REB kullanımı, MİB yükü, GİB güç tüketimi ve toplam güç tüketimi parametreleri göz önünde bulundurulmuştur.

Yörünge hatasının tespit edilebilmesi için MYH parametresi ele alınmıştır. MYH parametresinin tespiti için ise açık kaynaklarda yer alan evo aracı kullanılmıştır.

REB kullanımı, MİB kullanımı, GİB güç tüketimi ve toplam güç tüketimi parametreleri için NVIDIA Jetson TX1 donanımının sağladığı tegrastats yazılım arayüzü

kullanılmıştır. Tegrastats arayüzünün çıktı olarak verdiği .txt uzantılı metin dosyası ise MATLAB’da ayrıştırılarak analiz sonuçları görselleştirilmiştir.

Genel olarak değerlendirildiğinde, bulgular dokulu (textured) yüzeylere sahip ortamlarda öznitelik tabanlı yöntemlerin doğrudan yöntemlere göre MYH, MİB yükü, REB yükü ve toplam güç tüketimi açısından daha kullanışlı olduğuna işaret etmektedir.

Sonuçlara bakıldığında MYH açısından officeroom0 video dizisi haricinde tüm video dizileri üzerinde en etkin yöntemin ORB-SLAM2 olduğu gözlenmiştir. officeRoom0 veri kümesinde en iyi sonucu ise DSO yöntemi vermiştir. Li vd. 2020’de de belirtildiği üzere, officeroom0 veri kümesinde ORB-SLAM2 yönteminin yeterli öznitelik tespit edememesinin bir sonucu olarak MYH performansının daha düşük olduğu görülmüştür.

Döngü kapama özelliği sonucu gelen yüksek veri birikimi ile doğrudan ilişkili olarak, beklendiği üzere döngü kapama özelliğine sahip olmayan DSO yöntemi REB kullanımı bakımından en iyi sonucu vermiştir. REB kullanımı açısından en iyi yöntem DSO iken, REB kısıtlarını en çok zorlayan yöntem ise DDSO yöntemi olmuştur. DDSO yönteminin işlem sırasında NVIDIA Jetson TX1 donanımının REB kısıtını zorlamasının yanında, takas hafızasını da daha fazla kullanma eğilimi gösterdiği belirlenmiştir.

MİB yükü açısından sonuçlar incelendiğinde ICL-NUIM veri kümesi üzerinde DSO yöntemi daha iyi sonuçlar veriyorken, KITTI veri kümesi üzerinde ORB-SLAM2 yöntemi daha iyi sonuçlar vermiştir. Bununla birlikte EUROC veri kümesi üzerinde MH01 ve V102 için DSO yöntemi daha iyi iken, V203 veri kümesi üzerinde ORB-SLAM2 yöntemi daha iyi sonuç vermiştir. DSO ve DDSO yöntemlerinin karşılaştırıldığı TUM TGO veri kümesi üzerinde ise DSO yöntemi MİB yükü açısından daha iyi sonuçlar vermiştir. Bu haliyle, özniteliklerin daha az belirgin olduğu ICL-NUIM gibi ortamlarda ORB-SLAM2 yöntemi çözüm üretmek için daha fazla MİB yükü oluşturuyorken, özniteliklerin yoğun olduğu ve takip edilebildiği ortamlarda (KITTI – 00, 03, 07) ise ORB-SLAM2 yönteminin daha az MİB yükü oluşturduğu belirlenmiştir.

DSO yöntemi çözüm üretirken, NVIDIA Jetson TX1 donanımının GİB güç tüketimi, diğer yöntemlerdekine kıyasla daha yüksek olmuştur. GİB güç tüketimi açısından en iyi sonuç ORB-SLAM2 yöntemi ile elde edilmiştir. Bu sonuca işleme alınan çerçeve (keyframe) sayısındaki farkın sebep olduğu değerlendirilmektedir. DSO yönteminde

anahtar çerçeve sayısı saniyede 5-7 adet video çerçevesi iken (Engel vd. 2017), ORB-SLAM2’de her bir anahtar çerçeve arasında en az 20 video çerçevesi bulunmaktadır (Mur-Artal vd. 2015). Ek olarak DDSO yönteminde saniyedeki anahtar çerçeve sayısı saniyede 5-10 olarak verilmektedir (Gao vd. 2018).

Toplam güç tüketimleri, GİB güç tüketimleri sebebiyle, MİB yüklerinden farklılık göstermiştir. DSO yönteminin en az MİB yükünü getirdiği EUROC veri kümesinde, en az güç tüketimini ORB-SLAM2 sağlamıştır. livingRoom0 video dizisi haricindeki tüm video dizilerinde en iyi güç tüketimi performansını ORB-SLAM2 vermiştir. TUM TGO veri kümesi üzerinde ise toplam güç tüketimi açısından en iyi sonucu tüm video dizilerinde DDSO yöntemi vermiştir.

Sonuç olarak DDSO yönteminin REB kısıtlarını zorlaması ve çözüm süresinin uzun olması sebebiyle özellikle geniş ortamlarda kullanılmasının uygun olmadığı görülmüştür. Öznitelik çıkarımının daha kolay olduğu dokulu ortamlarda ise ortamın büyüklüğüne ya da küçüklüğüne bakılmaksızın ORB-SLAM2 yönteminin NVIDIA Jetson TX1 donanımı kısıtlarında çözüm üretmek için DSO ve DDSO yöntemlerine kıyasla MYH sonuçları, işlem hızı, MİB yükü ve toplam güç tüketimi açısından daha uygun yöntem olduğu belirlenmiştir. Dokusuz, kısa mesafeli ve az dönüşlerin olduğu ortamlarda ise DSO yönteminin konumlama ve haritalama probleminin çözümünde kullanılmasının uygun olduğu tespit edilmiştir.

KAYNAKLAR

- Anonim. 2020. Web Sitesi: http://www.cvlibs.net/datasets/kitti/eval_odometry.php?source=post_page-----
----- Erişim tarihi: 15.11.2020
- Anonim. 2021. Web Sitesi: https://macrovector.com/vector-image/people-walking-robots-along-streets-city-vector-illustration-55531?utm_content= Erişim Tarihi: 21.02.2021
- Anonim. 2021. Web Sitesi: https://macrovector.com/vector-image/smart-city-technologies-isometric-composition-view-vector-illustration-58645?utm_content=search_head Erişim Tarihi: 21.02.2021
- Anonim. 2021. Web Sitesi: <https://www.mi.com/tr/mi-robot-vacuum-mop-essential> Erişim Tarihi: 09.06.2021
- Anonymus 2021, Web Sitesi: <https://mars.nasa.gov/mars2020/timeline/landing/watch-online/> Erişim Tarihi: 09.06.2021
- Anonymus 2021, Web Sitesi: <https://waymo.com/waymo-driver/> Erişim Tarihi: 09.06.2021
- Mur-Artal, R., Tardos, J.D., Montiel, JMM, Galvez-Lopez, D 2016. ORB-SLAM2. Web Sitesi: https://github.com/raulmur/ORB_SLAM2 Erişim Tarihi: 10.17.2020
- Ball, D., Heath, S., Wiles, J., Wyeth, G., Corke, P. ve Milford, M. 2013. OpenRatSLAM: An open source brain-based SLAM system. *Autonomous Robots*, 34(3), 149-176. DOI: 10.1007/s10514-012-9317-9
- Bloesch, M., Omari, S., Hutter, M. ve Siegwart, R. 2015. Robust Visual Inertial Odometry Using A Direct EKF-Based Approach. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 298-304. <https://doi.org/10.3929/ethz-a-010566547>
- Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M. W. ve Siegwart, R. 2016. The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10), 1157-1163. <https://doi.org/10.1177/0278364915620033>
- Butler, D. J., Wulff, J., Stanley, G. B. Ve Black, M. J. 2012. A Naturalistic Open Source Movie For Optical Flow Evaluation. European Conference on Computer Vision, October 611-625, Berlin. DOI: [10.1007/978-3-642-33783-3_44](https://doi.org/10.1007/978-3-642-33783-3_44)
- Buyval, A., Afanasyev, I. ve Magid, E. 2017. Comparative Analysis of ROS-Based Monocular SLAM Methods for Indoor Navigation. Ninth International Conference on Machine Vision (ICMV 2016), Vol. 10341, 103411K, 17 Mart 2017, Fransa. <https://doi.org/10.1117/12.2268809>
- Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M., ve Tardós, J. D. 2021. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. IEEE Transactions on Robotics, DOI: 10.1109/TRO.2021.3075644

- Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I. ve Leonard, J. J. 2016. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6), 1309-1332. DOI: [10.1109/TRO.2016.2624754](https://doi.org/10.1109/TRO.2016.2624754)
- Concha, A. ve Civera, J. 2015. DPPTAM: Dense Piecewise Planar Tracking and Mapping from a Monocular Sequence. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS15)*, 28 Eylül – 2 Ekim 2015, 5686- 5693, Hamburg, Almanya
- Chong, T. J., Tang, X. J., Leng, C. H., Yogeswaran, M., Ng, O. E. ve Chong, Y. Z. 2015. Sensor technologies and simultaneous localization and mapping (SLAM). *Procedia Computer Science*, 76, 174-179. DOI: [10.1016/j.procs.2015.12.336](https://doi.org/10.1016/j.procs.2015.12.336)
- Delmerico, J. ve Scaramuzza, D. 2018. A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots. *International Conference on Robotics and Automation (ICRA)*, 21-26 Mayıs 2018, 2502-2509, Avustralya DOI: [10.1109/ICRA.2018.8460664](https://doi.org/10.1109/ICRA.2018.8460664)
- Durrant-Whyte, H. ve Bailey, T. 2006. Simultaneous localisation and mapping (SLAM): Part I the essential algorithms. *Robotics & Automation Magazine* 13(2), 99-110. DOI: [10.1109/MRA.2006.1638022](https://doi.org/10.1109/MRA.2006.1638022)
- Engel, J., Koltun, V. ve Cremers, D. 2017. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3), 611-625. DOI: [10.1109/TPAMI.2017.2658577](https://doi.org/10.1109/TPAMI.2017.2658577)
- Engel, J., Schöps, T. ve Cremers, D. 2014. LSD-SLAM: Large-Scale Direct Monocular SLAM. *European Conference on Computer Vision*, 6-12 Eylül 2014, 834-849 İsviçre. DOI: [10.1007/978-3-319-10605-2_54](https://doi.org/10.1007/978-3-319-10605-2_54)
- Engel, J., Usenko, V. ve Cremers, D. 2016. A Photometrically Calibrated Benchmark For Monocular Visual Odometry. *Computer Vision and Pattern Recognition*. 27-30 Haziran 2016, ABD.
- Filipenko, M. ve Afanasyev, I. 2018. Comparison of Various Slam Systems for Mobile Robot in an Indoor Environment. *International Conference on Intelligent Systems (IS)*, 25-27 Eylül 2018, 400-407. DOI: [10.1109/IS.2018.8710464](https://doi.org/10.1109/IS.2018.8710464)
- Forster, C., Carlone, L., Dellaert, F. ve Scaramuzza, D. 2016. On-manifold preintegration for real-time visual--inertial odometry. *IEEE Transactions on Robotics*, 33(1), 1-21. DOI: [10.1109/TRO.2016.2597321](https://doi.org/10.1109/TRO.2016.2597321)
- Forster, C., Pizzoli, M. ve Scaramuzza, D. 2014. SVO: Fast Semi-Direct Monocular Visual Odometry. *IEEE International Conference on Robotics and Automation (ICRA)*, 31 Mayıs-5 Haziran 2015, 15-22, Çin. DOI: [10.1109/ICRA.2014.6906584](https://doi.org/10.1109/ICRA.2014.6906584)
- Fuentes-Pacheco, J., Ruiz-Ascencio, J. ve Rendón-Mancha, J. M. 2015. Visual simultaneous localization and mapping: A survey. *Artificial Intelligence Review*, 43(1), 55-81. DOI: [10.1007/s10462-012-9365-8](https://doi.org/10.1007/s10462-012-9365-8)

- Gao, X., Wang, R., Demmel, N. ve Cremers, D. 2018. LDSO: Direct Sparse Odometry With Loop Closure. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1-5 Ekim 2018, 2198-2204, İspanya. <https://doi.org/10.1177/0278364913491297>
- Geiger, A., Lenz, P., Stiller, C. ve Urtasun, R. 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11), 1231-1237. DOI: 10.1177/0278364913491297
- Grisetti, G., Kümmerle, R., Stachniss, C. ve Burgard, W. 2010. A tutorial on graph-based SLAM. *Intelligent Transportation Systems Magazine*, 2(4), 31-43. DOI: [10.1109/MITS.2010.939925](https://doi.org/10.1109/MITS.2010.939925)
- Grupp, M. 2017. evo: Python package for the evaluation of odometry and SLAM. Web Sitesi: <https://github.com/MichaelGrupp/evo> Erişim tarihi: 14.08.2020
- Güçlü, O. 2018. RGB-D algılayıcılar kullanılarak eş zamanlı konum belirleme ve haritalama. Doktora tezi (basılmamış). Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, Ankara.
- Handa, A., Whelan, T., McDonald, J. ve Davison, A. J. 2014. A Benchmark For RGB-D Visual Odometry, 3D Reconstruction and SLAM. IEEE International Conference on Robotics and Automation (ICRA), 31 Mayıs-5 Haziran 2014, 1524-1531, Çin. DOI: [10.1109/ICRA.2014.6907054](https://doi.org/10.1109/ICRA.2014.6907054)
- Huletski, A., Kartashov, D. ve Krinkin, K. 2015. Evaluation of the Modern Visual Slam Methods. Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference (AINL-ISMW FRUCT), 9-14 Kasım 2015, 19-25, Rusya. DOI: [10.1109/AINL-ISMW-FRUCT.2015.7382963](https://doi.org/10.1109/AINL-ISMW-FRUCT.2015.7382963)
- Ibragimov, I. Z. ve Afanasyev, I. M. 2017. Comparison of ROS-Based Visual SLAM Methods in Homogeneous Indoor Environment. 14th Workshop on Positioning, Navigation and Communications (WPNC), 25-26 Ekim 2017, 1-6. ABD. DOI: [10.1109/WPNC.2017.8250081](https://doi.org/10.1109/WPNC.2017.8250081)
- İlci, V. ve Toth, C. (2020). High definition 3D map creation using GNSS/İMU/LiDAR sensor integration to support autonomous vehicle navigation. *Sensors*, 20(3), 899. DOI: [10.3390/s20030899](https://doi.org/10.3390/s20030899)
- Klein, G. ve Murray, D. 2007. Parallel Tracking and Mapping for Small AR Workspaces. 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 13-16 Kasım 2007, 225-234, Japonya. DOI: [10.1109/ISMAR.2007.4538852](https://doi.org/10.1109/ISMAR.2007.4538852)
- Korkmaz, M., Durdu, A. ve Tusun, Y. E. 2018. Sensor comparison for a real-time slam application. *International Journal of Information and Electronics Engineering*, 8(1), 1-4. DOI: [10.18178/IJIEE.2018.8.1.684](https://doi.org/10.18178/IJIEE.2018.8.1.684)
- Labbe, M. ve Michaud, F. 2014. Online Global Loop Closure Detection for Large-Scale Multi-Session Graph-Based SLAM. IEEE/RSJ International Conference on Intelligent Robots and Systems, 14-18 Eylül, 2661-2666, ABD. DOI: [10.1109/IROS.2014.6942926](https://doi.org/10.1109/IROS.2014.6942926)

- Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R. ve Furgale, P. 2015. Keyframe-based visual-inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3), 314-334. DOI: [10.1177/0278364914554813](https://doi.org/10.1177/0278364914554813)
- Li, Y., Brasch, N., Wang, Y., Navab, N., & Tombari, F. (2020). Structure-SLAM: Low-drift monocular SLAM in indoor environments. *IEEE Robotics and Automation Letters*, 5(4), 6583-6590. DOI: [10.1109/LRA.2020.3015456](https://doi.org/10.1109/LRA.2020.3015456)
- Liang, X., Chen, H., Li, Y. ve Liu, Y. 2016. Visual laser-slam in large-scale indoor environments. International Conference on Robotics and Biomimetics, 3-7 Aralık 2016, 19-24, Çin. DOI: [10.1109/ROBIO.2016.7866271](https://doi.org/10.1109/ROBIO.2016.7866271)
- Lin, K. H. ve Wang, C. C. 2010. Stereo-Based Simultaneous Localization, Mapping and Moving Object Tracking. IEEE/RSJ International Conference on Intelligent Robots and Systems, 18-22 Ekim 2010, 3975-3980, Tayvan. DOI: [10.1109/IROS.2010.5649653](https://doi.org/10.1109/IROS.2010.5649653)
- Lynen, S., Achtelek, M. W., Weiss, S., Chli, M. ve Siegwart, R. 2013. A Robust and Modular Multi-Sensor Fusion Approach Applied to Mav Navigation. IEEE/RSJ International Conference on Intelligent Robots and Systems, 3-7 Kasım 2017, 3923-3929, Japonya. DOI: [10.1109/IROS.2013.6696917](https://doi.org/10.1109/IROS.2013.6696917)
- Merzlyakov, A., ve Macenski, S. 2021. A Comparison of Modern General-Purpose Visual SLAM Approaches. arXiv preprint arXiv:2107.07589.
- Migliore, D., Rigamonti, R., Marzorati, D., Matteucci, M. ve Sorrenti, D. G. 2009. Use a Single Camera for Simultaneous Localization and Mapping with Mobile Object Tracking in Dynamic Environments. ICRA Workshop On Safe Navigation in Open and Dynamic Environments: Application to Autonomous Vehicles, 12-17 Mayıs 2009, Japonya.
- Mingachev, E., Lavrenov, R., Tsoy, T., Matsuno, F., Svinin, M., Suthakorn, J. ve Magid, E. 2020. Comparison of ROS-Based Monocular Visual SLAM Methods: DSO, LDSO, ORB-SLAM2 and Dynaslam. International Conference on Interactive Collaborative Robotics, 7-9 Ekim 2020, 222-233, Rusya. DOI: [10.1007/978-3-030-60337-3_22](https://doi.org/10.1007/978-3-030-60337-3_22)
- Mourikis, A. I. ve Roumeliotis, S. I. 2007. A Multi-State Constraint Kalman Filter for Vision-Aided Inertial Navigation. Proceedings IEEE International Conference On Robotics and Automation, 10-14 Nisan 2007, 3565-3572, İtalya.
- Mur-Artal, R. ve Tardós, J. D. 2017. ORB-SLAM2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5), 1255-1262. DOI: [10.1109/TRO.2017.2705103](https://doi.org/10.1109/TRO.2017.2705103)
- Mur-Artal, R., Montiel, J. M. M. ve Tardos, J. D. 2015. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5), 1147-1163. DOI: [10.1109/TRO.2015.2463671](https://doi.org/10.1109/TRO.2015.2463671)
- Peris, M., Martull, S., Maki, A., Ohkawa, Y. ve Fukui, K. 2012. Towards a Simulation Driven Stereo Vision System. Proceedings Of The 21st International Conference On Pattern Recognition (ICPR2012), 11-15 Kasım 2012, 1038-1042, Japonya.

- Pire, T., Fischer, T., Castro, G., De Cristóforis, P., Civera, J. ve Berles, J. J. 2017. S-PTAM: Stereo parallel tracking and mapping. *Robotics and Autonomous Systems*, 93, 27-42. <http://dx.doi.org/10.1016/j.robot.2017.03.019>
- Pizzoli, M., Forster, C. ve Scaramuzza, D. 2014. REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time. IEEE International Conference on Robotics and Automation (ICRA), 31 Mayıs- 5 Haziran, 2014, 2609-2616, Çin. DOI: [10.1109/ICRA.2014.6907233](https://doi.org/10.1109/ICRA.2014.6907233)
- Qin, T., Li, P. ve Shen, S. (2018). Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4), 1004-1020. DOI: [10.1109/TRO.2018.2853729](https://doi.org/10.1109/TRO.2018.2853729)
- Silberman, N., Hoiem, D., Kohli, P. ve Fergus, R. 2012. Indoor Segmentation And Support Inference From Rgb-d Images. European Conference On Computer Vision, 7-13 Ekim 2012, 746-760, İtalya. DOI: [10.1007/978-3-642-33715-4_54](https://doi.org/10.1007/978-3-642-33715-4_54)
- Sturm, J., Engelhard, N., Endres, F., Burgard, W. ve Cremers, D. 2012. A Benchmark for the Evaluation of RGB-D SLAM Systems. International Conference on Intelligent Robots and Systems, 7-12 Ekim 2012, 573-580, Portekiz. DOI: [10.1109/IROS.2012.6385773](https://doi.org/10.1109/IROS.2012.6385773)
- Sumikura, S., Shibuya, M., ve Sakurada, K. 2019. Openvslam: a versatile visual slam framework. In Proceedings of the 27th ACM International Conference on Multimedia, Ekim 2019, 2292-2295, DOI: <https://doi.org/10.1145/3343031.3350539>
- Thrun, S., Montemerlo, M., Koller, D., Wegbreit, B., Nieto, J. ve Nebot, E. 2004. FastSLAM: An efficient solution to the simultaneous localization and mapping problem with unknown data association. *Journal of Machine Learning Research*, 4(3), 380-407.
- Wang, C. C., Thorpe, C., Thrun, S., Hebert, M. ve Durrant-Whyte, H. 2007. Simultaneous localization, mapping and moving object tracking. *The International Journal of Robotics Research*, 26(9), 889-916. <https://doi.org/10.1177/0278364907081229>
- Wangsiripitak, S. ve Murray, D. W. 2009. Avoiding Moving Outliers in Visual SLAM by Tracking Moving Objects. IEEE International Conference on Robotics and Automation, 12-17 Mayıs 2009, 375-380, Japonya. DOI: [10.1109/ROBOT.2009.5152290](https://doi.org/10.1109/ROBOT.2009.5152290)
- Yousif, K., Bab-Hadiashar, A. ve Hoseinnezhad, R. 2015. An overview to visual odometry and visual SLAM: Applications to mobile robotics. *Intelligent Industrial Systems*, 1(4), 289-311. DOI: [10.1007/s40903-015-0032-7](https://doi.org/10.1007/s40903-015-0032-7)
- Zikos, N. ve Petridis, V. 2011. 6-DoF low dimensionality SLAM (l-SLAM). *Journal of Intelligent & Robotic Systems*, 79(1), 55-72. DOI [10.1007/s10846-014-0029-6](https://doi.org/10.1007/s10846-014-0029-6)