T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GNSS KARIŞTIRMALI ORTAMLARDA MUTLAK GÖRSEL KONUMLANDIRMA DESTEKLİ GÖRSEL-ATALETSEL NAVİGASYON YAKLAŞIMI

Fatih BAŞATEMUR

YÜKSEK LİSANS TEZİ
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Programı

Danışman
Dr. Öğr. Üyesi Erkan USLU

Kasım, 2025

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GNSS KARIŞTIRMALI ORTAMLARDA MUTLAK GÖRSEL KONUMLANDIRMA DESTEKLİ GÖRSEL-ATALETSEL NAVİGASYON YAKLAŞIMI

Fatih BAŞATEMUR tarafından hazırlanan tez çalışması 28.11.2025 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Dr. Öğr. Üyesi Erkan USLU
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Dr. Öğr. Üyesi Erkan USLU, Danışman
Yıldız Teknik Üniversitesi _____

Doç. Dr. Hamza Osman İLHAN, Üye
Yıldız Teknik Üniversitesi _____

Dr. Ögr. Üyesi Nihal ALTUNTAŞ, Üye
İstanbul Üniversitesi-Cerrahpaşa _____

Danışmanım Dr. Öğr. Üyesi Erkan USLU sorumluluğunda tarafımca hazırlanan GNSS KARIŞTIRMALI ORTAMLARDA MUTLAK GÖRSEL KONUMLANDIRMA DESTEKLİ GÖRSEL-ATALETSEL NAVİGASYON YAKLAŞIMI başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Fatih BAŞATEMUR

İmza

Aileme ve en iyi arkadaşıma
ithaf ediyorum.

TEŞEKKÜR

Bu tez çalışmasının tamamlanmasında bana destek olan ve katkı sağlayan birçok kişiye teşekkür etmek istiyorum.

Öncelikle, tez danışmanım Dr. Öğr. Üyesi Erkan USLU'a sonsuz teşekkürlerimi sunarım. Kendisi, tez süresince sağladığı değerli yönlendirmeleri, sabrı ve teşvik edici yaklaşımları ile bana her zaman ilham kaynağı olmuştur.

Bu süreçte bana her zaman moral veren ve yanımda olan aileme ve arkadaşlarıma da teşekkür etmek istiyorum.

Fatih BAŞATEMUR

İÇİNDEKİLER

SİMGE LİSTESİ vii

KISALTMA LİSTESİ xi

ŞEKİL LİSTESİ xiii

TABLO LİSTESİ xiv

ÖZET xv

ABSTRACT xvii

1 GİRİŞ 1

1.1 Giriş . . . 1
1.2 Problem Tanımı . . . 2
1.3 Tezin Amaç ve Kapsamı . . . 2
1.4 Tezin Genel Organizasyonu . . . 3

2 LİTERATÜR İNCELEMESİ 5

3 METODOLOJİ 8

3.1 Bilgisayarlı Görü . . . 8
3.1.1 Öznitelik Çıkarımı ve Eşleme . . . 8
3.1.2 Düzlem Dönüşümü Temelleri . . . 9
3.1.3 Pinhole Kamera Projeksiyon Modeli . . . 11
3.2 Görsel-Ataletsel Odometri . . . 12
3.2.1 Faktör Graf . . . 12
3.3 Faktör Graf ve Optimizasyon Eşdeğerliği . . . 16
3.3.1 Maksimum Olabilirlik Formülasyonu . . . 17
3.3.2 Negatif Logaritma ve Toplam Maliyet . . . 17
3.3.3 Gaussian Faktörler ve Ağırlıklı En Küçük Kareler . . . 17
3.4 Sparse Matris Temsili . . . 19
3.4.1 Faktör Grafiğinin Sparse Matris Temsili . . . 19
3.4.2 Optimizasyon Problemi ve Doğrusallaştırma Adımı . . . 20

3.4.3 Sparse Matris Avantajları 22
3.4.4 Ataletsel Sensör Ön Entegrasyonu 24
3.4.5 Durum Tahmini Optimizasyonunda Marjinalleştirme ve Kayan Pencereler 26
3.5 VINS-Fusion 29
3.5.1 VINS Global Optimizasyon 30
3.5.2 GNSS Verisinin Global Optimizasyon Grafiğine Entegrasyonu 33

4 YÖNTEM **37**
4.1 AVL ve VINS-Fusion Tabanlı Navigasyon Sistemi 37
4.1.1 AVL Entegrasyonu 37
4.1.2 AVL Konumunun Elde Edilmesi 40
4.2 Tezde Ele Alınan AVL Geliştirmeleri 41
4.2.1 Manyetometre Destekli Görsel Dönmelerin Telafisi 42
4.2.2 Görüş Alanına Bağlı Çözünürlük Dengelemesi 43
4.2.3 Yönelim Tabanlı Konum İyileştirmesi 45

5 DENEYSEL ÇALIŞMALAR VE SONUÇLAR **48**
5.1 Simülasyon Ortamı 48
5.1.1 Iris Hava Aracı Model Analizi 48
5.1.2 Gövde ve Kamera Arasındaki Kinematik Dönüşüm 49
5.1.3 Dönüşüm Metodolojisi 50
5.1.4 Rotor Bilgileri ve Uçuş Performansı 51
5.2 Uçuş Görev Planları 52
5.2.1 Üçgensel Form Uçuşu 52
5.2.2 Çembersel Form Uçuşu 53
5.3 Simülasyon Ortam Sonuçları 54
5.3.1 Üçgensel Form Uçuş Analizi 54
5.3.2 Çembersel Form Uçuş Analizi 55
5.4 Veri Seti Sonuçları 58
5.4.1 MARS-LVIG Veri Seti 58
5.4.2 Yöntem Değerlendirme ve Analiz 59

6 SONUÇ **65**

KAYNAKÇA **67**

TEZDEN ÜRETİLMİŞ YAYINLAR **71**

SİMGE LİSTESİ

$\mathbf{r}_{ij}$	Anahtar kareler arası hata vektörü
i, j	Anahtar kare indisleri
$\mathbf{r}_{\text{seq}}$	Ardışık kenarlar için hata terimi
$\mathbf{p}_{avl_gnss}^{enu}$	AVL ile elde edilen konum (ENU)
Σ_{avl_gnss}	AVL kovaryans matrisi
Lon_{avl}	AVL tarafından üretilen boylam
Lat_{avl}	AVL tarafından üretilen enlem
$\mathbf{I}$	Birim matris
λ	Boylam
Lon	Boylam (Longitude)
θ	Dikilme açısı (pitch) / Açı parametresi
d_E	Doğu (East) yönünde yer değiştirme (metre)
$\mathbf{r}_{\text{loop}}$	Döngü kapatma hata terimi
$\mathcal{L}$	Döngü kapatma kenarları kümesi
Σ_{loop}	Döngü kapatma kovaryans matrisi
$\mathbf{x}$	Durum vektörü / 2D nokta
R_e	Dünya yarıçapı (yaklaşık 6371 km)
Lat	Enlem (Latitude)
$\mathbf{p}_{gnss}^{enu}$	GNSS anteninin ENU koordinatlarındaki konumu
$\mathbf{p}_{b}^{gnss}$	GNSS anteninin gövdeye göre ötelemesi
Σ_{gnss}	GNSS kovaryans matrisi
W	Görüntü genişliği (piksel)
Lat_b	Görüntü kesitin sağ alt köşe enlemi

Lon_b	Görüntü kesitin sağ alt köşe boylamı
Lat_t	Görüntü kesitin sol üst köşe enlemi
Lon_t	Görüntü kesitin sol üst köşe boylamı
H	Görüntü yüksekliği (piksel) / İHA yerden yüksekliği (metre)
C_x, C_y	Görüntüdeki piksel koordinatları
$\mathcal{F}_b$	Gövde (body) koordinat sistemi
$\mathbf{C}_n^b$	Gövde çerçevesinden yerel seyrüsefer çerçevesine dönüşüm
$\mathbf{R}_{b_k}^w$	Gövde çerçevesinden dünya çerçevesine rotasyon matrisi
F_{body}	Gövde koordinat sistemi (alternatif gösterim)
$\mathbf{v}_b$	Gövde koordinatlarında gözlem vektörü
$\mathbf{p}_w^{b_k}$	Gövdenin dünya koordinatlarındaki konumu
$\mathbf{q}_w^{b_k}$	Gövdenin yönelim kuaternyonu
GSD_{map}	Harita Yer Örnekleme Mesafesi (m/piksel)
P_h	Harita üzerindeki alan genişliği (piksel)
P_v	Harita üzerindeki alan yüksekliği (piksel)
λ_a	Hava aracının boylamı
ϕ_a	Hava aracının enlemi
h_a	Hava aracının irtifası
λ_t	Hedef noktasının boylamı
ϕ_t	Hedef noktasının enlemi
h_t	Hedef noktasının rakımı
$\rho(\cdot)$	Huber normu fonksiyonu
θ_i	i anındaki dikilme açısı
ψ_i	i anındaki sapma açısı
ϕ_i	i anındaki yatış açısı
$\hat{\mathbf{p}}_{ij}^b$	i ve j anahtar kareleri arasında ölçülen göreceli pozisyon
$\hat{\psi}_{ij}$	i ve j arasında ölçülen göreceli sapma açısı
D_v	İHA tarafından taranan alanın dikey mesafesi (metre)
D_h	İHA tarafından taranan alanın yatay mesafesi (metre)

ψ_j	j anındaki sapma açısı
$\mathbf{p}_{b_k}^{enu}$	k anında gövdenin ENU konumu
$\mathbf{r}_{avl_gnss,k}$	k anındaki AVL hata terimi
$\mathbf{r}_{gnss,k}$	k anındaki GNSS hata terimi
$T_{\text{cam}}^{\text{body}}$	Kamera çerçevesinden gövde çerçevesine dönüşüm matrisi
$T_{\text{cam}}^{\text{mount}}$	Kamera çerçevesinden mount noktasına dönüşüm matrisi
F_{cam}	Kamera optik koordinat sistemi
FOV_v	Kameranın dikey görüş açısı
FOV_h	Kameranın yatay görüş açısı
d_N	Kuzey (North) yönünde yer değiştirme (metre)
F_{mount}	Mount noktası koordinat sistemi
$T_{\text{mount}}^{\text{body}}$	Mount noktasından gövde çerçevesine dönüşüm matrisi
$\mathcal{X}$	Optimize edilecek durum vektörleri kümesi
s	Ölçek faktörü
$\mathbf{t}$	Öteleme vektörü
$\mathcal{S}$	Poz grafiğindeki tüm sıralı kenarlar kümesi
$\mathbf{R}$	Rotasyon matrisi
ψ	Sapma açısı (yaw)
$\hat{\mathbf{p}}_{gnss}^{enu}$	Tahmin edilen GNSS anteni konumu (ENU)
$\hat{\mathbf{p}}_{b}^{enu}$	Tahmin edilen gövde konumu (ENU)
$\mathbf{p}_{b_k}^{w}$	VIO tarafından tahmin edilen gövde konumu
$\mathbf{R}_{w}^{enu}$	VIO dünya çerçevesinden ENU çerçevesine dönüşüm matrisi
$\mathbf{p}_{w}^{enu}$	VIO dünya çerçevesinin ENU konumu
W	VIO dünya koordinat çerçevesi
$\mathbf{R}_x(\phi)$	X ekseni etrafında dönüş matrisi
$\mathbf{R}_y(\theta)$	Y ekseni etrafında dönüş matrisi
ϕ	Yatış açısı (roll) / Enlem
h_{AGL}	Yerden yükseklik (Above Ground Level, metre)
$\mathcal{F}_n$	Yerel seyrüsefer koordinat sistemi

$\mathbf{v}_n$	Yerel seyrüsefer koordinatlarında gözlem vektörü
d_U	Yukarı (Up) yönünde yer değiştirme (metre)
$\mathbf{R}_z(\psi)$	Z ekseni etrafında dönüş matrisi
k	Zaman indeksi / Anahtar kare indeksi

KISALTMA LİSTESİ

AGL	Above Ground Level
ATE	Absolute Trajectory Error
AVL	Absolute Visual Localization
BA	Bundle Adjustment
BRIEF	Binary Robust Independent Elementary Features
CNN	Convolutional Neural Network
DBoW2	Bag of Binary Words
DoF	Degrees of Freedom
DSMAC	Digital Scene Matching Area Correlator
DSP	Digital Signal Processing
EKF	Extended Kalman Filter
ENU	East-North-Up
FAST	Features from Accelerated Segment Test
FoV	Field of View
FPS	Frames Per Second
GNSS	Global Navigation Satellite System
GSD	Ground Sample Distance
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
İHA	İnsansız Hava Aracı
LIO	LiDAR-Inertial Odometry
LLA	Latitude, Longitude, Altitude
MAP	Maximum-a-Posteriori

MSCKF	Multi-State Constraint Kalman Filter
NED	North, East, Down
NetVLAD	Network Vector of Locally Aggregated Descriptors
OpenVINS	Open Visual-Inertial System
ORB	Oriented FAST and Rotated BRIEF
PnP	Perspective-n-Point
PTAM	Parallel Tracking and Mapping
RF	Radio Frequency
RGB-D	Red Green Blue - Depth
RMSE	Root Mean Square Error
RTK	Real Time Kinematics
RVL	Relative Visual Localization
SLAM	Simultaneous Localization and Mapping
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
VINS	Visual-Inertial Navigation System
VIO	Visual-Inertial Odometry
VLAD	Vector of Locally Aggregated Descriptors
VO	Visual Odometry
VPR	Visual Place Recognition
VSLAM	Visual Simultaneous Localization and Mapping
WGS84	World Geodetic System 1984

ŞEKİL LİSTESİ

Şekil 3.1 Pinhole kamera projeksiyon modeli 11
Şekil 3.2 Beş değişkenli g fonksiyonunun faktör grafiği 14
Şekil 3.3 Kalman filtresi adımları temsili faktör grafiği 16
Şekil 3.4 Şekil 3.3 faktör grafiğinin L sparse matris gösterimi 20
Şekil 3.5 Marginalize işleminin faktör graf uygulanması 26
Şekil 3.6 x_m değişken düğümünün marjinalleştirilmesi 27
Şekil 3.7 Görsel-Ataletsel Odometri için temsili faktör grafiği 29
Şekil 3.8 VINS-Mono Global Optimizasyon süreci 31
Şekil 4.1 Referans hava aracı görseli ile temsili eşleme alanı [49] 40
Şekil 5.1 Simülasyon ortamında tanımlanan Iris çok rotorlu hava aracı . . . 48
Şekil 5.2 Kamera ve Body Mount eksen takımları arasındaki dönüşüm . . . 50
Şekil 5.3 Üçgensel form uçuş planı 52
Şekil 5.4 Çembersel form uçuş planı 53
Şekil 5.5 100m üçgensel form uçuşu 54
Şekil 5.6 100m çembersel form uçuşu 55
Şekil 5.7 VINS-Fusion (döngü kapatma aktif) – 100 m çembersel uçuş . . . 57
Şekil 5.8 Rural town uçuş senaryosunun taranan bölge görseli 58
Şekil 5.9 Valley uçuş senaryosunun taranan bölge görseli 59
Şekil 5.10 Sabit ve değişken yaw stratejilerinin, AMvalley03 veri seti üzerindeki eşleştirme süreklilik profilleri 59
Şekil 5.11 Temel ve önerilen konum iyileştirmesinin AMtown03 veri seti üzerinde değerlendirmesi 61
Şekil 5.12 AMtown03 uçuş senaryosunda VINS-Fusion, VINS-Fusion + AVL ve AVL çalışmalarından elde edilen trajektori sonuçları . . . 62
Şekil 5.13 AMvalley03 uçuş senaryosunda VINS-Fusion, VINS-Fusion + AVL ve AVL çalışmalarından elde edilen trajektori sonuçları . . . 63

TABLO LİSTESİ

Tablo 5.1 Uçuş planları özet tablosu 53
Tablo 5.2 VINS-Fusion + AVL çalışma başarımının, referans VINS-Fusion algoritması ile simülasyon uçuş sekansları üzerindeki karşılaştırmalı sonuçları 56
Tablo 5.3 VINS-Fusion + AVL çalışma başarımının, referans VINS-Fusion algoritması ile *100m_iris_circle* uçuş sekansı üzerindeki karşılaştırma sonuçları 57
Tablo 5.4 Temel ve önerilen yöntemlerin mutlak yörünge hatası karşılaştırması 61
Tablo 5.5 Geliştirilen VINS-Fusion + AVL çalışmasının, referans VINS-Fusion ve FAST-LIO2 algoritması ile MARS-LVIG veri seti üzerindeki sonuçları 63

ÖZET

GNSS Karıştırmalı Ortamlarda Mutlak Görsel Konumlandırma Destekli Görsel-Ataletsel Navigasyon Yaklaşımı

Fatih BAŞATEMUR

Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Dr. Öğr. Üyesi Erkan USLU

Günümüzde insansız hava aracı (İHA) navigasyon sistemlerinin büyük bir kısmı Küresel Navigasyon Uydu Sistemi (GNSS) teknolojisine bağımlıdır. Ancak GNSS sinyallerinin karıştırılması, engellenmesi veya kısıtlı erişim gibi durumlarda güvenilmez hale gelmesi, seyrüsefer kabiliyetini kritik düzeyde etkilemektedir. Bu nedenle, İHA'ların harici referans sistemlerine ihtiyaç duymaksızın otonom konum belirlemesi büyük önem arz etmektedir. Bu tez çalışmasının temel amacı, GNSS'siz ortamlarda ve yüksek irtifa uçuşlarında, görsel odometri yaklaşımlarını kullanarak İHA'lar için güvenilir ve sürekli bir pozisyon verisi sağlamaktır.

Bu problem alanına yönelik olarak, kamera ve ataletsel ölçüm birimi (IMU) gibi hafif sensörleri etkin bir şekilde kullanan Görsel-Ataletsel Odometri (VIO) ve Görsel Eşzamanlı Konumlandırma ve Haritalama (VSLAM) yöntemleri öne çıkmaktadır. Çalışmamızda, pozisyon tahmini için referans olarak VINS-Fusion algoritması temel alınmıştır. Yapılan analizler, temel VINS-Fusion algoritmasının kısa menzilli ve düz rotalarda yüksek doğruluk sergilemesine karşın, uzun menzilli ve yüksek irtifalı görevlerde kümülatif hata birikimi (sürüklenme) nedeniyle performansının önemli ölçüde düştüğünü göstermiştir.

Bu sınırlılığı gidermek amacıyla, VINS-Fusion algoritmasının global optimizasyon sürecine Mutlak Görsel Konumlandırma (Absolute Visual Localization, AVL) verilerinin entegre edildiği yeni bir yaklaşım önerilmiştir. Bu yöntemde, AVL sisteminden elde edilen mutlak konum kısıtları, VINS-Fusion'ın faktör grafına dahil

edilerek sistemin uzun vadedeki doğruluğu ve kararlılığı artırılmıştır. Önerilen sistemin performansı, farklı irtifalarda tasarlanan düz hat, eğimli ve kapalı döngü uçuş senaryolarında, mutlak yörünge hatası (ATE) ve döngü kapatma metrikleri kullanılarak kapsamlı bir şekilde değerlendirilmiştir.

Simülasyon ve veri seti tabanlı testler sonucunda, geliştirilen entegre yaklaşımın temel VINS-Fusion algoritmasına kıyasla konumlandırma doğruluğunu yaklaşık %83 oranında iyileştirdiği tespit edilmiştir. Bu iyileşme sayesinde, özellikle yüksek irtifa ve uzun menzilli uçuşlarda gözlemlenen sürüklenme etkisi başarılı bir şekilde azaltılmış ve daha tutarlı yörünge tahminleri elde edilmiştir. Elde edilen bulgular, görsel odometri tabanlı sistemlerin mutlak konum verileriyle desteklenmesinin, GNSS'nin mevcut veya güvenilir olmadığı zorlu koşullarda dahi İHA'lar için kararlı ve hassas bir seyrüsefer çözümü sunabileceğini kanıtlamaktadır.

Anahtar Kelimeler: Görsel-Ataletsel Odometri, Görsel Konumlandırma

ABSTRACT

Visual-Inertial Navigation Approach Enhanced By Absolute Visual Localization In GNSS-Denied Environments

Fatih BAŞATEMUR

Department of Computer Science
Master of Science Thesis

Supervisor: Asst. Prof. Erkan USLU

In recent years, a large portion of unmanned aerial vehicle (UAV) navigation systems has relied on Global Navigation Satellite System (GNSS) technology. However, GNSS signals may become unreliable in cases of interference, blockage, or limited access, critically affecting navigation capability. Therefore, enabling UAVs to determine their position autonomously without depending on external reference systems is of great importance. The primary objective of this thesis is to provide a reliable and continuous position estimation for UAVs in GNSS-denied environments and high-altitude flight scenarios using visual odometry techniques.

To address this problem domain, Visual-Inertial Odometry (VIO) and Visual Simultaneous Localization and Mapping (VSLAM) methods, which efficiently utilize lightweight sensors such as cameras and inertial measurement units (IMUs), have gained prominence. In this study, the VINS-Fusion algorithm is adopted as the reference framework for position estimation. Analyses indicate that while the baseline VINS-Fusion algorithm performs accurately in short-range and straight-path trajectories, its performance significantly degrades in long-range and high-altitude missions due to cumulative drift errors.

To overcome this limitation, a novel approach is proposed by integrating Absolute Visual Localization (AVL) data into the global optimization process of VINS-Fusion. In this method, absolute position constraints derived from the AVL system are incorporated into the factor graph of VINS-Fusion, enhancing the

long-term accuracy and stability of the system.

Simulation and dataset-based experiments demonstrate that the proposed integrated approach improves positioning accuracy by approximately 83% compared to the baseline VINS-Fusion algorithm. This improvement effectively mitigates drift effects observed during long-range and high-altitude flights, yielding more consistent trajectory estimations. The results confirm that augmenting visual odometry systems with absolute position information can provide robust and precise navigation solutions for UAVs, even under GNSS-denied or degraded conditions.

Keywords: Visual-Inertial Odometry, Visual Localization

1
GİRİŞ

1.1 Giriş

Günümüzde İnsansız Hava Araçları (İHA'lar), çok çeşitli senaryolarda ve ticari uygulamalarda yaygın bir şekilde kullanılmaktadır. Bu araçlarda tercih edilen navigasyon sistemleri genellikle, Küresel Navigasyon Uydu Sistemleri (GNSS), Ataletsel Navigasyon Sistemleri (INS) veya daha hassas konum verisi için GNSS ve RTK kombinasyonlarından oluşmaktadır. GNSS, jeosantrik bir referans çerçevesi kullanarak mutlak lokalizasyon sağlar ve birikimli hatalardan etkilenmez, bu da onu bugün elimizdeki en güvenilir navigasyon sistemi haline getirir. Ancak, GNSS sistemleri sahtecilik (spoofing), bastırma (jamming) ve radyo frekans girişimi (RF interference) durumlarında işlevsiz kalabilmektedir. Diğer taraftan, INS harici bir referansa bağlı olmadığı için bağımsız bir çözüm sunar; ancak ivme ve açısal hızdaki küçük ölçüm hatalarının birikmesi ve ortaya çıkan kaymalar nedeniyle zaman içinde konum hatası oluşturur. RTK destekli GNSS sistemleri ise GNSS Bayes hatalarının Taşıyıcı Faz Düzeltmesi (Carrier-Phase Correction) yöntemi ile metrelik hataların santimetre hassasiyetine düşürülmesini sağlamaktadır; ancak maliyet ve bakım gereksinimlerine ek olarak GNSS donanımının gereksinimi, bu sistemi de spoofing ortamlarında etkisiz kılmaktadır.

Küresel Navigasyon Uydu Sistemleri'nin erişilemediği veya güvenilir olmadığı ortamlarda, kesintisiz pozisyon bilgisinin sağlanması kritik bir öneme sahiptir. Bu bağlamda, monoküler kamera ile ataletsel ölçüm birimlerinin (Inertial Measurement Units, IMU) entegre bir şekilde kullanıldığı Görsel-Ataletsel Odometri (VIO) yöntemleri geliştirilmiştir. Ancak, VIO sistemlerinde, bağıl görsel konumlandırma (Relative Visual Localization, RVL) prensibiyle çalıştıkları için, uzun süreli kullanımlarda birikimli hatalar (drift) veya ıraksama (divergence) meydana gelebilir.

Bu çalışmada, mevcut görsel-ataletsel navigasyon sistemlerine (Visual-Inertial Navigation System, VINS), mutlak görsel konumlandırma (Absolute Visual

Localization, AVL) çıktılarının füzyon edilmesiyle sistemin zorlu koşullarda daha güvenilir sonuçlar üretmesi hedeflenmiştir. Geliştirilen yöntem, oluşturulan simülasyon ortamı ve kullanılan gerçek yaşam verikümeleri üzerinde GNSS karıştırmalı ortamlar için kaymasız ve kararlı konum bilgisi sağlayabildiği gözlemlenmiştir.

1.2 Problem Tanımı

İnsansız hava araçlarının GNSS'e bağımlı olması, bu sistemlerin spoofing, jamming ve radyo frekans girişimi gibi durumlarda görev yapamamasına neden olmaktadır. Alternatif olarak kullanılan ataletsel navigasyon sistemleri, zamanla birikimli hata üretmektedir. Bu durum, özellikle uzun süreli ve GNSS erişiminin mümkün olmadığı görevlerde güvenilir konumlandırma yapılmasını engellemektedir.

Son yıllarda geliştirilen Görsel-Ataletsel Odometri sistemleri, IMU ve kamera verilerini birleştirerek bağıl konum tahmini yapmaktadır. Ancak bu sistemlerde, dış referansa bağlı kalmadan yapılan konumlandırma, birikimli hata sorununu tamamen ortadan kaldıramamaktadır.

1.3 Tezin Amaç ve Kapsamı

Bu çalışmanın temel amacı, GNSS sinyallerinin erişilemediği veya güvenilir olmadığı ortamlarda İnsansız Hava Araçlarının uzun süreli görevlerde kararlı ve kaymasız konum bilgisi elde edebilmesi için Mutlak Görsel Konumlandırma (AVL) çıktılarının VINS-Fusion algoritması ile entegrasyonunu gerçekleştirmektir. Geleneksel Görsel-Ataletsel Odometri sistemlerinin zamanla birikimli hata (drift) üretmesi ve mutlak konumlandırma yeteneğinin olmaması, uzun mesafeli görevlerde ciddi güvenilirlik sorunlarına yol açmaktadır.

Bu çalışma kapsamında:

- AVL çıktılarının, VINS-Fusion'ın global optimizasyon sürecine AVL konumlandırma faktörü olarak entegrasyonu gerçekleştirilmiştir. AVL yalnızca enlem ve boylam (2-DoF) sağladığı için, irtifa bilgisi, barometrik irtifa tahminlerinden alınarak sistem tasarlanmıştır.
- Hava aracının anlık duruş bilgileri (attitude) ve AGL irtifası kullanılarak, AVL algoritmasının çıktısı olan hedef nokta koordinatlarından hava aracının gerçek konumunun türetilmesini sağlayan Yönelim Tabanlı Konum İyileştirme algoritması geliştirilmiştir.

- Gazebo simülasyon ortamında Iris çok rotorlu hava aracı modeli üzerinde, monoküler kamera ve IMU entegrasyonu ile gerçekçi test senaryoları oluşturulmuştur. Sensör kalibrasyonu ve koordinat dönüşümleri detaylı olarak analiz edilmiştir.
- Geliştirilen sistem, MARS-LVIG veri seti üzerinde test edilmiş ve VINS-Fusion, VINS-Fusion + AVL ve FAST-LIO2 algoritmaları ATE metriği üzerinden karşılaştırılmıştır.
- Hava aracının yönelim durumunun, SuperGlue tabanlı özellik eşleştirme algoritması üzerindeki etkileri deneysel olarak değerlendirilmiş, eşleme performansını arttırmak ve başarılı konum çıkarımını desteklemek için görüş alanına bağlı çözünürlük dengelemesi ifade edilmiştir.

Çalışmanın kapsam dışı bırakılan hususları şunlardır: çoklu sensör füzyonu (LiDAR, stereo kamera vb.), hesaplama yükü optimizasyonu ve yeni nesil özellik eşleştirme algoritmaları. Bu çalışma, önceden haritalanmış bölgelerde geçerli olup, harita güncelliği ve mevsimsel değişikliklere karşı hassasiyet konuları, gelecek çalışmalarda ele alınması gereken sınırlamalar olarak belirlenmiştir.

1.4 Tezin Genel Organizasyonu

Bu tez beş ana bölümden oluşmaktadır. İlk bölüm olan giriş bölümünde, problem tanımı, çalışmanın motivasyonu, amaç ve kapsamı ele alınmıştır.

Bölüm 2 - Literatür İncelemesi: Görsel konumlandırma ve görsel-ataletsel füzyon alanındaki gelişim süreci kronolojik olarak incelenmektedir. MSCKF, OpenVINS gibi filtre tabanlı yaklaşımlar, PTAM, ORB-SLAM serisi, VINS-Mono/Fusion gibi optimizasyon tabanlı SLAM ve VIO sistemleri ile GVINS, Wildnav gibi GNSS/AVL entegrasyonlu çalışmalar değerlendirilerek mevcut yöntemlerin güçlü ve zayıf yönleri tartışılmaktadır.

Bölüm 3 - Metodoloji: Çalışmanın teorik temelleri sunulmaktadır. Bilgisayarlı görü kavramları, öznitelik çıkarımı ve epipolar geometrisi matematiksel olarak açıklanmaktadır. VIO'nun çalışma prensibi, IMU ön-entegrasyonu, kayar pencere optimizasyonu ve marjinalizasyon teknikleri ele alınmaktadır. VINS-Fusion algoritmasının mimarisi, loop closure, poz grafiği optimizasyonu, 4-DoF yaklaşımı ve GNSS entegrasyon metodolojisi detaylandırılmaktadır.

Bölüm 4 - Yöntem: Tez kapsamında geliştirilen özgün katkılar sunulmaktadır. AVL çıktılarının VINS-Fusion'a global konumlandırma kaynağı olarak

entegrasyonu, görsel konumlandırma faktör uyarlaması, SuperGlue özellik eşleştirme algoritması ile Wildnav tabanlı AVL konum elde edilmesi ve Yönelim Tabanlı Konum İyileştirme algoritması detaylı olarak sunulmaktadır.

Bölüm 5 - Deneysel Çalışmalar ve Sonuçlar: Gazebo simülasyon ortamı ve Iris hava aracı modeli sunulmaktadır. Görsel dönme stratejilerinin eşleştirme performansına etkileri ve Yönelim Tabanlı Konum İyileştirme algoritmasının %50 oranında iyileşme sağladığı gösterilmektedir. MARS-LVIG veri seti üzerinde VINS-Fusion + AVL sisteminin performansı, VINS-Fusion ve FAST-LIO2 ile karşılaştırılmakta ve AMtown03/AMvalley03 sekanslarında sırasıyla %83/%38 hata azaltımı elde edildiği ortaya konmaktadır. Bölüm sonunda tezin genel değerlendirmesi yapılmakta ve gelecek çalışmalar önerilmektedir.

2
LİTERATÜR İNCELEMESİ

Görsel konumlandırmaya dayalı ilk örneklerden biri, 1980'li yıllarda geliştirilen DSMAC (Digital Scene Matching Area Correlator) sistemidir [1]. Bu sistem, Tomahawk Konvansiyonel Seyir Füzesi'nin (cruise missile) hedefi daha hassas vurabilmesi için anlık konum bilgisini düzenlemektedir.

1997 yılında, Mars yüzeyinde görev yapan rover için konumlandırma amacıyla stereo kameralar, iniş sırasında çekilen yüksek çözünürlüklü görüntüler ve hareket tabanlı fark analizi kullanılmıştır. Bu yöntemle altı serbestlik derecesinde (6-DoF) konum ve yönelim kestirimi yapılmıştır [2]. Böylece Mars yüzeyine inen ilk gezici araçta bu teknolojiler başarıyla kullanılarak VO'nun potansiyeli ortaya konmuştur.

Görüntü ve ataletsel veri füzyonu alanında Mourikis ve Roumeliotis (2006), filtre tabanlı bir yöntem olan MSCKF (Multi-State Constraint Kalman Filter) algoritmasını önermiştir [3]. Bu yaklaşım, 3B özellik konumlarını açıkça tahmin etmeksizin, çoklu kamera pozisyonlarından gözlemlenen geometrik kısıtları kullanarak hesaplama yükünü azaltmaktadır. Bu çalışma, hesaplama maliyeti açısından görüntü destekli navigasyonda referans çalışma olmuştur. Fast-MSCKF [4], MSCKF'nin hesaplama maliyetini %80 azaltabilmiş ve konum tahmini doğruluğunu %20 artıran bir çalışma olmuştur. Yine OpenVINS [5] kayar pencere (sliding window) formülasyonu ve modüler kovaryans yönetimi ile esnek bir altyapı sağlamıştır.

PTAM, izleme (tracking) ve haritalama (mapping) işlemlerini ayrı iş parçacıklarında gerçekleştirerek gerçek zamanlı tek kamerayla SLAM uygulamaları için temel bir altyapı sunmuş, Bundle Adjustment tabanlı optimizasyonu popülerleştirmiştir [6]. Harita ve kamera pozlarını optimize etmek için anahtar kareler üzerinde Bundle Adjustment (BA) kullanması, doğruluk açısından filtre tabanlı yöntemleri geride bırakmıştır.

Tek kameralı SLAM sistemlerinde, görüntüden doğrudan derinlik bilgisi elde

edilemediği için harita başlatma (initialization) süreci kritik öneme sahiptir. ORB-SLAM [7], PTAM'ın temel mimarisini korurken, otomatik başlatma, covisibility grafikleri ve DBoW2 [8] tabanlı yer tanıma gibi yenilikler ekleyerek monocular SLAM alanında önemli bir dönüm noktası olmuştur. Otomatik harita başlatma (Automatic Map Initialization), bu amaçla homografi ve fundamental matrisini paralel olarak hesaplar ve ardışık görüntüler arası paralaks efekt olduğunda başlatma yapar. Aksi durumda, yeterli paralaks olmadığında, başlatmayı geciktirerek hatalı harita oluşumunu önler. Ayrıca, survival of the fittest yaklaşımı, harita noktaları ve anahtar karelerin dinamik olarak seçilip elenmesi, hem kompakt bir harita oluşturmayı hem de uzun süreli çalışma (lifelong operation) yeteneğini mümkün kılar. Bu strateji, gereksiz veri birikimini önlerken, zorlu keşif senaryolarında bile kararlılığı korur.

ORB-SLAM2' de [9], stereo ve RGB-D kameralar için kapsamlı destek eklemiştir. Bu genişletme, monocular sistemin ölçek belirsizliği, harita başlatma zorlukları ve ölçek kayması gibi sınırlamalarını ortadan kaldırmıştır. Ayrıca, ORB-SLAM2'de Full BA (tüm harita ve anahtar karelerin optimizasyonu) döngü kapatma sonrasına entegre edilerek küresel tutarlılık daha da güçlendirilmiştir.

VINS-Mono [10], farklı olarak VIO çalışması olarak ön plana çıkmaktadır. Monocular kamera ve IMU kombinasyonu ile sınırlı kaynaklarda yüksek performans sunarken hareket halinde bile sistemin başlatılabilmesine, kamera - IMU dışsal (extrinsic) parametrelerinin online kalibrasyonuna ve drift problemini engellemek için loop closure yeteneklerine sahiptir.

VINS-Mono çalışmasının bir uzantısı olan VINS-Fusion [11], stereo kamera ve GPS gibi çoklu sensör türünün dahil edilmesi ile optimizasyon tabanlı çoklu sensör durum tahmincisidir. Global optimizasyona dahil edilen GPS faktörü ile outdoor ortamda drift problemini çözerek alternatif GNSS kaynakları ile GNSS' siz ortamlar için aday çalışma olmaktadır.

ORB-SLAM3 [12], ORB-SLAM2'nin temel altyapısını genişleterek visual-inertial SLAM, çoklu harita yönetimi ve geliştirilmiş yer tanıma (visual place recognition) gibi kritik yenilikler sunmaktadır. Öncelikle, monocular, stereo ve RGB-D kameralarla IMU verilerini MAP (Maximum-a-Posteriori) tabanlı optimizasyonla entegre ederek monocular sistemlerdeki ölçek belirsizliğini ortadan kaldırıp, düşük dokulu ortamlarda kararlılığı artırır. Atlas sistemi ile takip kaybı durumunda yeni haritalar oluşturup bunları geliştirilmiş yer tanıma ile birleştirerek uzun süreli operasyonlarda süreklilik sağlar. Ayrıca, geliştirilmiş yer tanıma algoritması DBoW2'ye eklenen üçlü kovizyon keyframe doğrulaması ile recall oranını

yükseltir, döngü kapatma ve harita birleştirme süreçlerini hızlandırır.

GVINS [13], GNSS ölçümlerini, görsel ve ataletsel verileri sıkı bir entegrasyonla birleştirerek kaymasız ve küresel tutarlı konum tahmini sağlar. Ayrıca coarse-to-fine başlatma mekanizmasıyla, GNSS-VIO konum dönüşümlerini dinamik olarak kalibre ederek global pozisyonu hassaslaştırır. GNSS ham ölçümlerine sıkı şekilde bağlı bir çalışma olması, görsel GNSS kaynaklı çalışmalarda implementasyon dezavantajı oluşturmaktadır.

NetVLAD, CNN' den yararlanarak görüntü özniteliklerini, VLAD (Vector of Locally Aggregated Descriptors) vektörel temsiline yorumlayarak görsel yer tanıma (Visual Place Recognition, VPR) çalışmalarında kullanılmaktadır [14]. Yine günümüzde, GNSS'siz ortamlar için AVL alanında yapılan çalışmalardan Wildnav, İHA tarafından elde edilen görüntü özniteliklerini coğrafi referanslı uydu görüntülerinden oluşan bir veritabanıyla eşleştirerek görsel temelli mutlak lokalizasyon gerçekleştirmektedir [15]. Bu yaklaşımlar mutlak konumlandırma için yararlı gözükmekte; ancak mevsimsel farklılık, nehir ya da insan yerleşiminin farklılaşması kaynaklı eşleşme yapılamadığı durumlar için sürekli çözüm sağlayamamaktadır.

3
METODOLOJİ

Bu bölümde, tezin kavramsal temelini oluşturan bilgisayarlı görü, Görsel-Ataletsel Odometri teorisi ve VINS-Fusion yaklaşımı ayrıntılı olarak ele alınmıştır.

3.1 Bilgisayarlı Görü

İnsanlar, çevrelerindeki üç boyutlu dünyayı algılamak için görsel sistemi etkili ve hızlı bir şekilde kullanırlar; bilgisayarlı görü araştırmaları da aynı amaca yönelik gelişmiş matematiksel modeller ve algoritmalar üretir.

Bilgisayarlı görü araştırmalarının temel hedefi, görüntülerde yer alan dünyayı ayrıntılı bir şekilde modellemek ve bu dünyanın geometrik yapısını, ışıklandırma koşullarını ve renk dağılımını anlamlandırmaktır [16].

3.1.1 Öznitelik Çıkarımı ve Eşleme

Özellik algılama ve eşleştirme, bilgisayarlı görme alanında uzun süredir temel bir işlem adımı olarak yer almaktadır. Başlangıçta stereo eşleştirme gibi problemlerin çözümünde kullanılan bu yöntemler, zamanla görüntü birleştirme uygulamalarında ve tamamen otomatik üç boyutlu modelleme süreçlerinde önemli bir rol üstlenmiştir. Görsel verilerdeki belirgin yapısal öğelerin tespit edilmesi ve bu öğeler arasındaki eşleşmelerin sağlanması, özellikle Çoklu Perspektiften Sahne Üretimi (Multi-View Scene Reconstruction) ile sahnenin yeniden inşası çalışmalarında merkezi bir öneme sahiptir. Gelişen algoritmalar sayesinde bu süreçler daha güvenilir ve otomatik hale gelmiş; böylece görsel odometri, haritalama ve konumlandırma gibi pek çok uygulamanın temelini oluşturmuştur.

Özellik noktalarını bulmak ve bunların eşleşmelerini belirlemek için iki ana yaklaşım vardır. İlk yaklaşım, bir görüntüdeki özellikleri bulup, bunları korelasyon veya en küçük kareler gibi yerel arama teknikleri kullanarak hassas bir şekilde takip etmektir. İkinci yaklaşım ise, ele alınan tüm görüntülerde bağımsız olarak

özellikleri tespit etmek ve ardından bu özellikleri yerel görünümlerine dayanarak eşleştirmektir. İlk yaklaşım, görüntülerin yakın görüş noktalarından alındığı veya hızlı ardışıklıkla çekildiği durumlar (örneğin, video dizileri) için daha uygundur. İkinci yaklaşım ise, büyük miktarda hareket veya görünüm değişikliğinin beklendiği durumlar için daha uygundur, örneğin panoramaların birleştirilmesinde ya da geniş tabanlı stereo eşleşmelerin oluşturulmasında kullanılmıştır [16].

3.1.2 Düzlem Dönüşümü Temelleri

2B dönüşümler, geometrik şekillerin veya noktaların düzlemdeki pozisyonlarını, şekillerini ve yönelimlerini değiştirmek için kullanılan temel araçlardır. Bu bölümde, bu dönüşümlerin matematiksel temelleri ve çeşitleri ele alınmaktadır. En basit dönüşümler, Şekil 2.4'te gösterildiği gibi 2B düzlemde gerçekleşir.

2B düzlemde x noktasına öteleme şu şekilde ifade edilir:

$$\mathbf{x}' = \mathbf{x} + \mathbf{t} \tag{3.1}$$

temsil şu şekilde genişletilebilir:

$$\mathbf{x}' = \begin{bmatrix} \mathbf{I} & \mathbf{t} \end{bmatrix} \tilde{\mathbf{x}} \tag{3.2}$$

$$\mathbf{x}' = \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \tilde{\mathbf{x}} \tag{3.3}$$

Bir 2×3 matris kullanmak, daha kompakt bir gösterimle sonuçlanır. Ayrıca, 2×3 matrise $[\mathbf{0}^\top \ 1]$ satırını ekleyerek elde edilen tam sıralı 3×3 bir matris kullanmak, matris çarpımını kullanarak dönüşümleri zincirleme ve ters dönüşümleri hesaplama imkanı sağlar. Homojen bir vektör olan $\tilde{\mathbf{x}}$ ile $\mathbf{x}$'in her iki tarafta yer aldığı herhangi bir denklemde, bu her zaman $\mathbf{x}$'in yerine konabilir [17].

Bu dönüşüm aynı zamanda 2B rijit gövde hareketi veya 2B Öklid dönüşümü olarak bilinir (çünkü Öklid mesafeleri korunur). Şu şekilde yazılabilir:

$$\mathbf{x}' = \mathbf{R}\mathbf{x} + \mathbf{t} \tag{3.4}$$

$$\mathbf{x}' = \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \tilde{\mathbf{x}} \tag{3.5}$$

burada R matrisi,

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{3.6}$$

bir ortonormal döndürme matrisidir ve $\mathbf{R}\mathbf{R}^\top = \mathbf{I}$ ve $|\mathbf{R}| = 1$ eşitlikleri sağlanır [17].

Ölçeklendirilmiş dönme, bir ölçek faktörü ve döndürme matrisinin birleşimi ile ifade edilen dönüşüm türüdür ve şu şekilde gösterilebilir:

$$\mathbf{x}' = s\mathbf{R}\mathbf{x} + \mathbf{t} \tag{3.7}$$

veya

$$\mathbf{x}' = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \end{bmatrix} \tilde{\mathbf{x}} \tag{3.8}$$

bu dönüşümde, $|\mathbf{R}| = 1$ eşitliği şartı ortadan kalkar çünkü dönüşüm matrisine bir ölçek faktörü s eklenir.

3.1.2.1 Affine Dönüşümü

Affine dönüşümü, doğrusal dönüşümlerin (örn. dönme, ölçekleme, kaydırma) ve ötelemenin bir birleşimidir. Bu dönüşüm, bir nesneyi bir uzayda başka bir pozisyona, şekle veya boyuta taşımak için kullanılır ve şu şekilde yazılır:

$$\mathbf{x}' = \mathbf{A}\tilde{\mathbf{x}} \tag{3.9}$$

$\mathbf{A}$, 2×3 boyutunda affine dönüşüm matrisi, doğrusal dönüşüm (ilk iki sütun) ile ötelemenin (son sütun) birleşimidir [17].

$$\mathbf{A} = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix} \tag{3.10}$$

Affine dönüşümü, bir doğruyu başka bir doğruya dönüştüresinden dolayı yansıtılan düzlem doğrultularını paralel tutar.

3.1.2.2 Perspektif Dönüşümü

Perspektif dönüşümü, homojen koordinatlar üzerinde işlem gerçekleştirir:

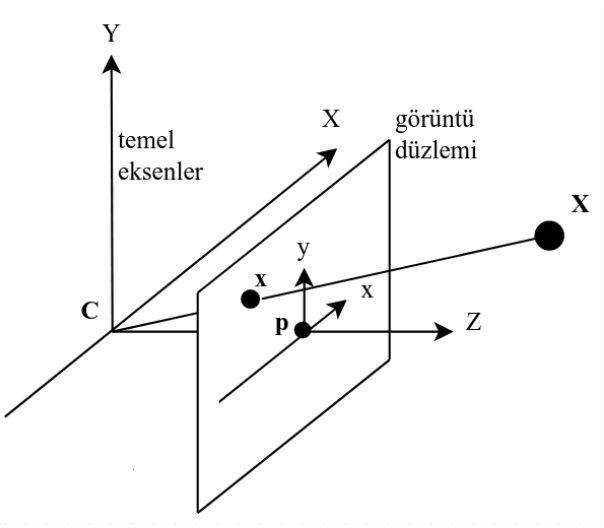
$$\tilde{\mathbf{x}}' = \mathbf{H}\tilde{\mathbf{x}} \tag{3.11}$$

$\mathbf{H}$, 3×3 boyutunda homojen matristir. Homojen dönüşümlerde, $\mathbf{H}$ yalnızca bir ölçek faktörüne kadar tanımlıdır. Bu, $\mathbf{H}$'nin herhangi bir skaler ile çarpıldığında aynı dönüşümü temsil ettiği anlamına gelir.

Dönüşüm sonrası elde edilen, ($\tilde{\mathbf{x}}'$) koordinatlarına, ölçek faktörü 1 olacak şekilde normalizasyon uygulanır. Dönüşüm sonucu $[x', y', w']^\top$ olarak elde edilen bir nokta, $[x'/w', y'/w', 1]^\top$ biçiminde normalleştirilir. Bu, dönüşümün ölçekten bağımsız olarak tutarlı bir şekilde yorumlanmasını sağlar [18]:

$$x' = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + h_{22}}, \quad y' = \frac{h_{10}x + h_{11}y + h_{12}}{h_{20}x + h_{21}y + h_{22}} \tag{3.12}$$

3.1.3 Pinhole Kamera Projeksiyon Modeli



Şekil 3.1 Pinhole kamera projeksiyon modeli

Pinhole kamera modeli, uzaydaki noktaların bir düzleme merkezi projeksiyonunu ele alınarak ifade edilebilir. Projeksiyon merkezi, Öklid koordinat sisteminin orijini

olarak alınır ve $Z = f$ düzlemi, görüntü düzlemi veya odak düzlemi olarak adlandırılır.

Pinhole kamera modeline göre, uzayda $\mathbf{X} = (X, Y, Z)^T$ koordinatlarına sahip bir nokta, bu noktadan projeksiyon merkezine çizilen doğrunun görüntü düzlemini kestiği noktaya eşlenir [18].

Şekil 3.1 üzerinde C, kamera merkezi; p ise ana nokta (principal point) olarak tanımlanır. Burada kamera merkezi, koordinat orijininde; görüntü düzlemi ise kamera merkezinin önünde yeralır.

3.2 Görsel-Ataletsel Odometri

VIO sistemleri, kamera ve IMU'dan elde edilen verileri bir araya getirerek zamana bağlı sapmaları büyük ölçüde azaltır. IMU'lar yüksek frekansta ivme ve dönüş bilgisi sağlarken, kameralar çevresel detayları zengin şekilde yakalayarak görsel bağlam oluşturur. Tek bir monoküler kamerayla yapılan SLAM uygulamalarında ölçek belirsizliği ortaya çıkarken, IMU eklemesi sayesinde (hareket dejenere olmadığı sürece) hem ölçek hem de 3B pozisyon tahminleri güvenilir biçimde elde edilebilir. Bu bütünleşik yaklaşım, yalnızca saf ataletsel veya saf görsel odometri yöntemlerine kıyasla daha dengeli, kararlı ve uzun süreli konum takibi sağlar [19].

3.2.1 Faktör Graf

Navigasyon sistemlerinde, farklı sensörlerden elde edilen verilerin etkin bir şekilde birleştirilmesi, sensör füzyonu, sistemin kararlı davranması için kritik rol oynamaktadır. Bir sistemin hareket dinamikleri, zaman içinde toplanan ölçümler ile entegre edilerek sistemin anlık durumu (konum, hız, yönelim vb.) başarılı bir şekilde tahmin edilebilir. Bu alanda, Rudolph E. Kalman'ın 1960'taki çalışmasının [20] ardından geliştirilen Kalman filtresi ve onun çeşitli türevleri, on yıllar boyunca sensör füzyonu için endüstri standardı haline gelmiştir. Bu filtrelerin ilk ve en çarpıcı başarılarından bazıları, Apollo görevleri gibi erken dönem uzay ve havacılık navigasyonunda ve GPS gibi Küresel Navigasyon Uydu Sistemleri verilerinin entegrasyonunda gözlemlenmiştir. Bu uygulamalar, gürültülü sensör verilerinden anlamlı navigasyon bilgisi çıkarmanın gücünü ortaya koymaktadır.

Teorik olarak, lineer Kalman filtresi, sistem dinamikleri ve ölçüm modelleri tamamen lineer olduğunda ve tüm gürültü kaynakları bilinen, Gauss dağılımlı ve beyaz gürültü özelliklerini taşıdığında, istatistiksel olarak optimal maksimum olabilirlik tahmincisini sunar. Ancak, gerçek dünya uygulamalarında karşılaşılan

sistemlerin büyük çoğunluğu (örneğin, robot kinematiği, yönelim temsilleri) bu katı lineerlik ve Gauss varsayımlarını karşılamaktan uzaktır. İşte bu pratik zorluklar, lineer olmayan sistemlere uyum sağlamak üzere tasarlanmış Genişletilmiş Kalman Filtresi (Extended Kalman Filter, EKF) [21], Kokusuz Kalman Filtresi (Unscented Kalman Filter, UKF) [22] gibi çok sayıda Kalman filtresi varyantının geliştirilmesine ve yaygınlaşmasına neden olmuştur.

Son yıllarda ise, sensör ve sistem dinamiklerinin yüksek derecede lineer olmayan doğası karşısında, sensör verilerini birleştirmek için faktör grafiği tabanlı optimizasyon yaklaşımları öne çıkmıştır. Faktör grafiklerinin genel bir grafiksel model olduğu Sum-Product algoritması altında ilk kez net ve kapsamlı bir şekilde Kschischang ve arkadaşları [23] tarafından ortaya konmuş, yine aynı dönemde DSP alanındaki uygulamaları ise Loeliger [24] tarafından derlenmiştir.

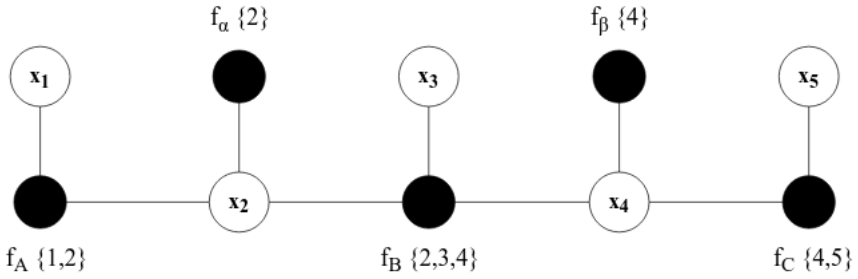
Mobil robotik literatüründe R. Chalita ve arkadaşı öncü olarak robot-merkezli koordinat sistemi ile dünya-merkezli koordinat sistemi arasındaki ilişkiyi formel olarak tanımlayarak pozisyon referanslamayı netleştirmiş; bu sayede robotun konum ve yönelimini, gürültülü ölçümler altında bile tahmin edilebilecek bir problem olarak formüle etmiştir. Ayrıca, döngü kapanışı (loop closing) kavramını içeren tutarlı bir dünya modellemesi önererek, SLAM güncel çalışmaların temel fikrini ortaya koymuştur [25]. Lazer sensörlerinde birikimli hata (drift) sorununa çözüm olarak, SLAM problemini bir graf optimizasyon problemi olarak formüle eden Feng Lu ve arkadaşı [26] bu çalışma ile faktör grafiği tabanlı modern SLAM algoritmalarının temelini atmıştır.

Robotik alanına Square Root SAM [27], GraphSLAM [28] gibi çalışmalarla giriş yapan faktör grafikleri, doğası temelinde, lineer olmayan optimizasyon problemlerini etkin bir şekilde formüle edip çözebilme, aykırı ölçüm verilerini (outliers) daha sağlam bir şekilde ele alabilme ve SO(3) gibi kısıtlı geometrik uzaylarda (manifoldlar) doğrudan optimizasyon yapabilme yeteneğine sahiptir. Bu yetenekler, modern navigasyon ve SLAM sistemlerinde çevresel ve ataletsel ölçüm dinamiklerine karşı avantaj sağlamaktadır [29].

Faktör grafiği, bir global fonksiyonun birçok değişkene bağlı olarak alabileceği değerleri, yerel fonksiyonların çarpımı biçiminde göstermeyi sağlayan; gizli değişkenler (latent/hidden variables) ve faktörler adında iki tür düğümden oluşan, iki parçalı (bipartite) bir grafiktir.

$$g(x_1, x_2, x_3, x_4, x_5) = f_A(x_1, x_2)\, f_B(x_2, x_3, x_4)\, f_C(x_4, x_5)\, f_\alpha(x_2)\, f_\beta(x_4) \quad (3.13)$$

Frey ve arkadaşları tarafından ortaya atılan bu model de [30], g global fonksiyon, $f_A, f_B, f_C, f_\alpha, f_\beta$ ise yerel fonksiyonlardır ve her biri yalnızca g'nin tüm değişkenleri arasından belirli alt kümelere bağımlı olarak 3.13 denklemi ile ifade edilir.



Şekil 3.2 Beş değişkenli g fonksiyonunun faktör grafiği

Boş daireler değişken düğümlerini (variable nodes), dolu daireler yerel fonksiyon düğümlerini (subset nodes) gösterir. Bir değişken düğümü ile bir fonksiyon düğümü arasında kenar varsa, o değişken o fonksiyonun bağımlı olduğu argümanlardan biridir. Genel olarak, $X = \{x_i\}_{i \in N}$ olsun; burada X, $N = \{1, 2, 3, \ldots, n\}$ sonlu bir küme tarafından indekslenen bir değişkenler topluluğudur. E, N'nin boş olmayan bir alt kümesi ise, X_E, X'in E tarafından indekslenen alt kümesini ifade eder [30]. Bir değişken x_i fonksiyonun argümanları arasında ise, x_i ile bir yerel fonksiyon düğümü f_E arasında kenar bulunur.

Başka bir deyişle, bir faktör grafiği, pek çok değişkenli bir fonksiyonun argümanlarını alt problemler hâline bölerek bu alt problemleri ayrı fonksiyonlarda toplar. Frey ve arkadaşlarının ana amacı, bu küresel fonksiyona ilişkin marjinal değerleri dağıtık mesajlaşma (message–passing) yoluyla hızlandırarak hesaplamaktır [30].

$$x_0 = \gamma + \xi \tag{3.14}$$

3.14 denklemi, EKF sisteminin başlangıç durumunun ($\mathbf{x}_0$) bir önsel tahmini (γ) ve bu tahmindeki belirsizliği veya gürültüyü ($\boldsymbol{\xi}$) tanımlar. Bu, filtrenin başlaması için gerekli olan ilk durum bilgisini sağlar [31].

$$z_k = h(x_k) + \eta \tag{3.15}$$

$$h_k(x_k) \approx h_k(\hat{x}_k) + H_k \cdot (x_k - \hat{x}_k) \tag{3.16}$$

$$H_k := \left.\frac{\partial h_k(x_k)}{\partial x_k}\right|_{x_k=\hat{x}_k} \tag{3.17}$$

3.15 denklem, bir ölçümün ($\mathbf{z}_k$) sistemin gerçek durumu ($\mathbf{x}_k$) ile nasıl ilişkili olduğunu tanımlar. $h()$ doğrusal olmayan ölçüm fonksiyonudur ve $\boldsymbol{\eta}$ ise ölçümdeki rastgele hatayı (measurement noise) temsil ederek, doğrusal olmayan ölçüm modelini ifade eder [31].

3.16, doğrusal olmayan $h()$ fonksiyonunun mevcut durum tahmini ($\hat{\mathbf{x}}_k$) etrafında birinci dereceden Taylor serisi açılımı ile doğrusallaştırılmasıdır [31].

3.17, ölçüm fonksiyonunun (h) duruma ($\mathbf{x}_k$) göre jakobiyen matrisinin tanımıdır. EKF'nin güncelleme adımında Kalman kazancını ve hata kovaryansını hesaplamak için bu matris kullanılır [31].

$$x_{k+1} = f(x_k, u_k) + \nu \tag{3.18}$$

$$f(x_k, u_k) \approx f(\hat{x}_k, u_k) + F \cdot (x_k - \hat{x}_k) \tag{3.19}$$

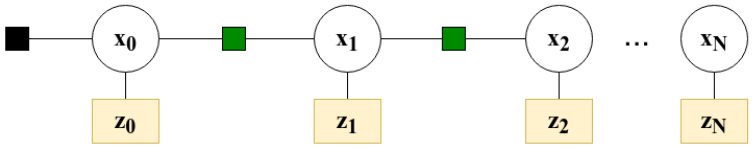
$$F := \left.\frac{\partial f(x_k, u_k)}{\partial x_k}\right|_{x_k=\hat{x}_k} \tag{3.20}$$

3.18 denklem, sistemin durumunun ($\mathbf{x}_k$) bir sonraki zaman adımında ($\mathbf{x}_{k+1}$) nasıl değiştiğini tanımlar. $f(\cdot, \cdot)$ doğrusal olmayan durum geçiş fonksiyonudur, $\mathbf{u}_k$ kontrol girdisidir ve $\mathbf{v}$ ise sistemdeki rastgele süreci (process noise) temsil eder. Bu denklem, doğrusal olmayan sistemin dinamik modelini verir.

3.19, doğrusal olmayan $f()$ fonksiyonunun mevcut durum tahmini ($\hat{\mathbf{x}}_k$) etrafında birinci dereceden Taylor serisi açılımı ile doğrusallaştırılmasıdır. EKF, doğrusal olmayan fonksiyonlarla doğrudan çalışmak yerine bu doğrusallaştırılmış versiyonları kullanır.

3.20, durum geçiş fonksiyonunun (f) duruma ($\mathbf{x}_k$) göre jakobiyen matrisinin tanımıdır. EKF'nin yayılım (prediction) adımında hata kovaryansını güncellemek için bu matris kullanılır.

Navigasyonda, faktörler, tıpki EKF' deki gibi, tipik olarak ölçümleri veya dinamikleri temsil eder. Grafiksel olarak, gizli değişkenler içi boş dairelerle, faktörler ise içi dolu dikdörtgenlerle gösterilir. Şekil 3.3' de siyah faktör, önceki (prior) değeri; yeşil faktör, x_i' den x_{i+1} durumuna dinamik geçişi temsil eden geçiş faktörü; sarı faktör düğümleri, ölçüm faktörü; x_i, gizli değişken düğümleridir.



Şekil 3.3 Kalman filtresi adımları temsili faktör grafiği

Grafikteki kenarlar yalnızca gizli değişken ve faktör düğümleri arasında bulunur. Gizli değişken düğümleri, tahmin edilmekte olan değişkenleri temsil eder ve yalnızca faktör düğümlerine bağlanabilir. Benzer şekilde, gizli değişkenlerin fonksiyonlarını temsil eden faktör düğümleri de yalnızca gizli değişken düğümlerine bağlanabilir (yani, faktörler başka faktörlerin fonksiyonu olamaz).

Bir faktör grafiğindeki her faktörle ilişkilendirilmiş üç öğe bulunur: gizli değişkenlerin bir fonksiyonu, o faktör için bir ölçüm değeri ve gizli değişkenin mevcut değerleri verildiğinde ölçülen değerin olabilirlik (likelihood) değerini döndüren bir olasılık dağılım fonksiyonu (ODF).

Dolayısıyla, bir faktör grafiği, alınan ölçümler göz önüne alındığında, bu alınan ölçümler ile tahmin edilmekte olan durum değerleri (yani gizli değişkenler) arasındaki olasılıksal ilişkinin bir temsilidir.

3.3 Faktör Graf ve Optimizasyon Eşdeğerliği

Aşağıda, faktör graf temsili ile eşdeğer maksimum olabilirlik optimizasyonunun nasıl bir ağırlıklı en küçük kareler problemine indirgeneceği adım adım gösterilmektedir. İlk olarak faktör graf üzerinden tanımlanan olasılık ifadeleriyle maksimum olabilirlik formülasyonu verilecek, ardından bu ifadenin negatif logaritması alınarak Gauss faktörler için ağırlıklı en küçük kareler biçimine dönüşümü incelenecektir.

3.3.1 Maksimum Olabilirlik Formülasyonu

Faktör graf üzerindeki her bir faktör $f \in \mathcal{F}$, komşu değişkenler $\mathcal{N}(f)$ arasındaki ortak olasılık bilgisini temsil eder. Gizli değişkenler kümesi X için maksimum olabilirlik problemi şu biçimdedir:

$$\hat{X} = \arg\max_{X} \prod_{f \in \mathcal{F}} p\big(f \mid \mathcal{N}(f)\big) \tag{3.21}$$

Bu çarpım ifadesini almak, faktörlerden bağımsız olarak tüm olasılıkların en yüksek olduğu X'i aramak demektir.

3.3.2 Negatif Logaritma ve Toplam Maliyet

Çarpımların optimizasyonunu kolaylaştırmak için negatif logaritma alınır:

$$\hat{X} = \arg\min_{X} - \sum_{f \in \mathcal{F}} \ln p\big(f \mid \mathcal{N}(f)\big). \tag{3.22}$$

Bu dönüşüm, ürün yerine toplam terimlerin en küçüğü olacak X'i bulmamıza imkân tanır [29].

3.3.3 Gaussian Faktörler ve Ağırlıklı En Küçük Kareler

Eğer her bir faktör, değişkenler arasındaki hatayı Gauss dağılımı ile tarif ediyorsa (yani her faktörün olasılık yoğunluğu bir üssel kare formunda ise), negatif log-olabilirlik

$$-\ln p(\cdot) \;\propto\; \big((\text{ölçüm} - \text{model})\big)^2 \tag{3.23}$$

biçimine indirgenir ve toplamda bir *weighted least-squares* (ağırlıklı en küçük kareler) problemi ortaya çıkar. Üç tip terim ile temsil edilir:

$$\frac{1}{2} \left\| x_0 - \gamma \right\|_{\Sigma_0}^2 \tag{3.24}$$

Ölcül (prior) terimi, başlangıç değişkeni x_0 ile önceden tanımlı γ değeri arasındaki farkın, kovaryans Σ_0 ile ağırlıklı karesidir.

$$\frac{1}{2}\sum_{k=0}^{N}\| h_k(x_k) - z_k\|_R^2 \tag{3.25}$$

Ölçüm (measurement) teriminde, her $k.$ zamandaki veya noktadaki ölçüm z_k ile model fonksiyonu $h_k(x_k)$ arasındaki farkın, ölçüm gürültüsü kovaryansı R ile ağırlıklı karesidir.

$$\frac{1}{2}\sum_{i=1}^{N}\| f(x_{i-1},\, u_{i-1}) - x_i\|_Q^2 \tag{3.26}$$

Geçiş (dynamics) terimi, bir önceki durum x_{i-1} ve kontrol girişi u_{i-1}'den beklenen yeni durum $f(x_{i-1}, u_{i-1})$ ile gerçek durum x_i arasındaki farkın, gürültü kovaryansı Q ile ağırlıklı karesini temsil edilir. öylece tüm faktörler toplanarak bir ağırlıklı en küçük kareler optimizasyonu ortaya çıkar, denklem 3.26 şekilde yazılabilir.

$$\begin{aligned}\hat{X} = \arg\min_X \; & \underbrace{\tfrac{1}{2}\big\|x_0 - \gamma\big\|_{\Sigma_0}^2}_{\text{Öncül (prior)}} \\ & + \underbrace{\tfrac{1}{2}\sum_{k=0}^{N}\big\|h_k(x_k) - z_k\big\|_R^2}_{\text{Ölçüm (measurement)}} \\ & + \underbrace{\tfrac{1}{2}\sum_{i=1}^{N}\big\|f(x_{i-1}, u_{i-1}) - x_i\big\|_Q^2}_{\text{Geçiş (dynamics)}}\end{aligned} \tag{3.27}$$

Burada

$$\|v\|_\Sigma^2 \;=\; v^\top \Sigma^{-1} v, \tag{3.28}$$

ve Σ_0, R, Q sırasıyla öncül, ölçüm ve geçiş gürültüsü kovaryans matrisleridir.

3.4 Sparse Matris Temsili

Faktör grafiklerinde, problem boyutu arttıkça, yani kestirilecek değişkenlerin ve ölçümlerin sayısı arttıkça, bu optimizasyon problemlerinin çözümü için gereken hesaplama karmaşıklığı önemli ölçüde artar. Doğrusal olmayan en küçük kareler yöntemleri (örneğin, Gauss-Newton veya Levenberg-Marquardt algoritmaları), her iterasyonda genellikle büyük bir doğrusal sistemin çözülmesini gerektirir [17, 32]. Bu sistemin bilgi matrisi (veya Hessian matrisinin bir yaklaşımı olan $J^\top J$) genellikle çok büyüktür. Ancak, faktör grafiklerinin doğası gereği, her bir faktör (ölçüm) az sayıda (bir grup) gizli değişkene bağlıdır. Yerel bağlantı (Local Connectivity) olarak isimlendirilen bu özellik, her faktörün az sayıda değişkeni ilişkilendirmesi ile optimizasyonda kullanılan Jakobiyen matrisinde kendini gösterir. Bir faktörle ilişkili olmayan değişkenlere ait kısmi türevler sıfır olacağından, Jakobiyen yüksek oranda sıfır eleman içerir (Bu özellikten dolayı Sparse (Seyrek) olarak isimlendirilir). Sonuç olarak, bu matristen türetilen bilgi matrisi ($J^\top J$) de önemli ölçüde seyrek bir yapıya sahip olur [33, 34].

Seyrek matris gösterimi ve bu matrisler üzerinde işlem yapmak üzere tasarlanmış Sparse Cholesky Decomposition gibi teknikler, bu hesaplama yükünü önemli ölçüde azaltır [35]. Faktör grafiğinin topolojisi, yani gizli değişkenler ve faktörler arasındaki bağlantı yapısı, Sparse sistemin seyrekliğini doğrudan belirler [33]. Sparse matris tekniklerinden yararlanılarak, hem bellek gereksinimleri hem de çözüm süresi, yoğun matris yöntemlerine kıyasla birkaç kat mertebesinde azaltılabilir; bu da büyük ölçekli SLAM ve Bundle Adjustment problemlerinin pratikte çözülebilir olmasını sağlar [34, 36]. Bu nedenle, modern SLAM ve BA kütüphanelerinin (örneğin, GTSAM [33], g²o [34], Ceres Solver [36]) temelinde verimli seyrek doğrusal cebir çözücüleri yatmaktadır.

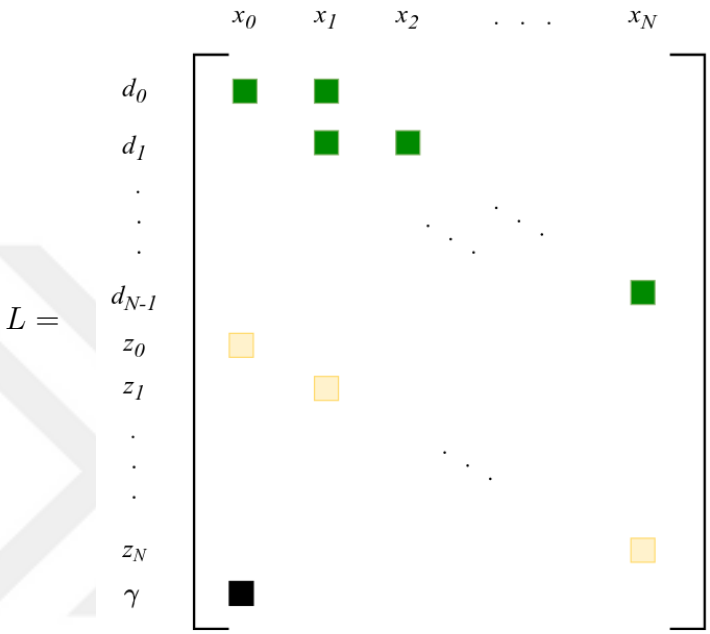
3.4.1 Faktör Grafiğinin Sparse Matris Temsili

Şekil 3.4'te, Şekil 3.3'te gösterilen iki parçalı faktör grafiği, L olarak adlandırılan bir matris aracılığıyla temsil edilmiştir. Bu matris, faktör grafiğinin temel topolojisini ve dolayısıyla optimizasyon probleminin yapısını yansıtır:

- L matrisinin her bir sütunu, faktör grafiğindeki bir gizli değişkene ($x_0, x_1, \ldots, x_N$) karşılık gelir.
- L matrisinin her bir satırı, grafikteki bir faktöre (ölçüm veya önsel bilgiye dayalı bir kısıta) karşılık gelir. Şekil 3.4'deki örnekte, d_0 ile etiketlenen satır, x_0 değişkeni üzerindeki bir önsel faktörü; $d_1, \ldots, d_{N-1}$ ile etiketlenen

satırlar ise genellikle ardışık gizli değişkenler (örneğin, x_{k-1} ve x_k) arasındaki hareket veya gözlem ölçümlerini (z_k) temsil eden faktörlere karşılık gelir.

- $L(i, j)$ matris elemanı, yalnızca i. faktör j. gizli değişkenle doğrudan ilişkiliyse (yani, faktör bu değişkene bağlıysa) sıfırdan farklı bir değer alır. Eğer bir faktör belirli bir değişkenle ilgili değilse, matrisin ilgili girişi sıfır olur.



Şekil 3.4 Şekil 3.3 faktör grafiğinin L sparse matris gösterimi

Faktör grafiklerinin doğasında bulunan yerel bağlantı (local connectivity) özelliği, L matrisinin seyrek (sparse) olmasını sağlar. Gerçek dünya problemlerinde, bir ölçüm (faktör) genellikle yalnızca az sayıda, çoğunlukla komşu olan gizli değişkenleri birbirine bağlar. Örneğin, bir robotun ardışık iki zaman adımındaki pozları arasındaki odometri ölçümü yalnızca bu iki poz değişkenini etkiler. Bu durum, L matrisinin büyük çoğunluğunun sıfır elemanlardan oluşmasına, sıfırdan farklı eleman sayısının ise toplam eleman sayısına kıyasla çok küçük kalmasına yol açar. Bu seyrek yapı, Şekil 3.3 matris görselleştirmesinde de açıkça görülmektedir.

3.4.2 Optimizasyon Problemi ve Doğrusallaştırma Adımı

Faktör grafikleri, gizli değişkenler arasındaki bağıntıları "faktör" adı verilen küçük potansiyeller aracılığıyla modeller ve tüm sistemin doğruluk ölçüsünü, denklem 3.26' de ifade edildiği gibi, ağırlıklı en küçük kareler (weighted least-squares) optimizasyon problemi üzerinden tanımlar. Amaç, tüm faktörler tarafından tanımlanan artıkların (residuals) karelerinin ağırlıklı toplamını minimize etmektir:

$$\min_{x} \sum_{k} \left\| h_k(x) - z_k \right\|_{R_k}^2 \tag{3.29}$$

Burada x optimize edilecek tüm gizli değişkenlerin vektörünü, $h_k(x)$ k. faktörün ilgili değişkenlere bağlı olarak beklenen değerini, z_k k. faktöre karşılık gelen gerçek ölçümü ve R_k ise k. faktörün bilgi matrisini (information matrix) temsil eder. Bu bilgi matrisi, her bir ölçümün optimizasyon üzerindeki ağırlığını belirler; daha güvenilir ölçümler (düşük kovaryans, yüksek bilgi) daha fazla ağırlığa sahip olur [29].

$h_k(x)$ fonksiyonu genellikle doğrusal olmadığından, bu tür problemler Gauss-Newton veya Levenberg-Marquardt gibi iteratif yöntemlerle çözülür. Bu yöntemlerin her bir iterasyonunda, problem mevcut değişken tahminleri $\hat{x}$ etrafında doğrusallaştırılır. k. faktör için $h_k(x)$ fonksiyonu, birinci dereceden Taylor serisi açılımı kullanılarak denklem 3.30 gibi ifade edilir:

$$h_k(x) \approx h_k(\hat{x}) + J_k \Delta x \tag{3.30}$$

Burada $\hat{x}$ değişkenlerin mevcut en iyi tahminini, $\Delta x = x - \hat{x}$ aranan düzeltme vektörünü ve J_k, h_k fonksiyonunun $\hat{x}$ noktasında ilgili değişkenlere göre hesaplanan Jakobiyan matrisini gösterir. Bu doğrusallaştırma sonucunda, k. faktörün ağırlıklı artık terimi şu şekle dönüşür:

$$\| h_k(\hat{x}) - z_k + J_k \Delta x_k \|_{R_k}^2 = \| y_k + J_k \Delta x_k \|_{R_k}^2 \tag{3.31}$$

Burada $y_k = h_k(\hat{x}) - z_k$ mevcut tahmine göre hesaplanan artık vektörüdür ve Δx_k yalnızca k. faktörün bağlı olduğu değişkenlere ait düzeltmeleri içeren alt vektördür.

Optimizasyon probleminin çözümü için, tüm faktörlerden gelen doğrusallaştırılmış ve ağırlıklandırılmış kareli hataların toplamının Δx'e göre türevi alınır ve sıfıra eşitlenir. Bu işlem, her bir faktörün optimizasyon problemine olan yerel katkısını ortaya çıkarır. Denklem 3.32, k. faktörden gelen terimin Δx_k'ye göre türevini göstermektedir:

$$\frac{\partial}{\partial \Delta x_k} \| y_k + J_k \Delta x_k \|_{R_k}^2 = 2 J_k^T R_k (y_k + J_k \Delta x_k) \tag{3.32}$$

Bu türev sıfıra eşitlenerek, 3.33 denklemi elde edilir:

$$J_k^T R_k J_k \Delta x_k = -J_k^T R_k y_k \tag{3.33}$$

$R^{-1} = S_R^T S_R$ ifadelendirmesi Denklem 3.33 için uygulandığında bir "karekök" bilgi matrisi S_R tanımlanır. Bu, ağırlıklandırmayı doğrudan Jakobyen'e dahil etmek için $J'_k = S_R J_k$ ve $y'_k = S_R y_k$ eşitlikleri kullanılır, böylece denklem 3.34 elde edilir.

$$(J'_k)^T J'_k \Delta x_k = -(J'_k)^T y'_k \tag{3.34}$$

Tüm faktörlerden gelen bu yerel katkılar (Hessian blokları ve gradyan blokları) uygun şekilde birleştirilerek, tüm değişkenler için tek bir büyük doğrusal denklem sistemi oluşturulur. Bu sisteme Normal Denklemler adı verilir ve Denklem 3.35 ile ifade edilir:

$$L^T L \Delta X = L^T Y \tag{3.35}$$

Burada, 3.35 denklemi $A\Delta X = b$ ifadelendirmesi ile daha genel temsil edilirse:

- ΔX: Tüm gizli değişkenler için global düzeltme vektörüdür.
- A (Denklem 3.35'daki $L^T L$'ye veya ağırlıklı durumda $J^T R J$'ye karşılık gelir): Sistemin yaklaşık Hessian matrisi olup minimize edilmeye çalışılmaktdır.
- b (Denklem 3.35'daki $L^T Y$'ye veya ağırlıklı durumda $-J^T R Y_{\text{full}}$'a karşılık gelir): Sistemin negatif gradyan vektörü olup sistemin mevcut durumdan optimuma doğru hangi yönde ve ne kadar "çekilmesi" gerektiğini ifade eder.

Denklem 3.35'daki L matrisi, tüm J_k Jakobyenlerinin uygun şekilde yığılmasıyla oluşan büyük bir Jakobyen matrisini, Y ise tüm y_k artık vektörlerinin yığılmasıyla oluşan büyük bir artık vektörünü temsil eder.

Her bir J_k Jakobyeni yalnızca k. faktörün bağlı olduğu değişkenlere karşılık gelen sütunlarda sıfırdan farklı elemanlara sahip olduğu için (yerel olması durumu) ve dolayısıyla genel sistem Jakobyeni L seyrek olduğu için, Hessian matrisi $A = L^T W L$ (W blok-diyagonal ağırlık matrisidir) de seyrektir. A matrisindeki sıfırdan farklı bir A_{ij} elemanı, yalnızca x_i ve x_j değişkenlerinin en az bir ortak faktör tarafından birbirine bağlanması durumunda ortaya çıkar. Bu, faktör grafiğinin komşuluk yapısını yansıtır [29].

3.4.3 Sparse Matris Avantajları

Normal denklemlerin ($A\Delta X = b$) çözümü, her optimizasyon iterasyonunun hesaplama açısından en maliyetli adımıdır. Eğer A matrisi N değişken için $N \times N$

boyutunda yoğun bir matris olsaydı, bu sistemi çözmek tipik olarak $O(N^3)$ zaman karmaşıklığına sahip olurdu. Bu, değişken sayısının artması durumunda çözüm süresini pratik olmayan seviyelere çıkarırdı.

Ancak, A matrisinin seyrek yapısı ile, bu lineer sistem çok daha verimli bir şekilde çözülebilir. Eigen [37], SuiteSparse [38] gibi özel olarak tasarlanmış seyrek lineer cebir kütüphaneleri, bu tür sistemleri çözmek için Cholesky ayrıştırması gibi gelişmiş linear cebir yaklaşımları kullanır. Faktör grafiğinin yapısına bağlı olarak, Normal denklemlerin çözümünün hesaplama karmaşıklığı, değişken sayısı N ile doğrusal ($O(N)$) veya doğrusala yakın ($O(N \log N)$ veya $O(N\sqrt{N})$) olabilir. Bu durum, çok büyük ölçekli optimizasyon problemlerinin bile makul hesaplama süreleri içinde çözülebilmesini mümkün kılar. Böylece hem olasılıksal tutarlılık korunur hem de büyük ölçekli problemler seyrek matris yapısından faydalanarak hesaplanabilir.

Bu çalışmada, konumlandırma problemi için MSCKF [3] gibi Kalman filtresi temelli yaklaşımların aksine; faktör grafiği tabanlı VIO yöntemleri tercih edilmiştir. Faktör grafiği optimizasyonunun, standart tek iterasyonlu Kalman filtrelerine kıyasla sunduğu bazı önemli avantajlar bu tercihin temelini oluşturmaktadır.

İlk olarak, faktör grafiği yaklaşımı, doğası gereği bir optimizasyon problemidir. Bu sayede, Kalman filtresinin her ölçüm adımında gerçekleştirdiği tek tahmin ve güncelleme döngüsünün aksine, bir maliyet fonksiyonunu minimize etmek amacıyla birden fazla iterasyon gerçekleştirir. Bu yapı, özellikle GNSS'siz ortamlarda görsel konumlandırma problemleri için daha doğru ve güvenilir durum tahminleri elde etme potansiyeli sunar [29].

İkinci olarak, faktör grafikleri, doğrusal olmayan ölçüm modellerinin ele alınmasında daha esnek bir yaklaşım sergiler. Standart EKF, ölçüm modelini mevcut durum tahmini etrafında yalnızca bir kez doğrusallaştırırken, faktör grafiği optimizasyonu her iterasyon adımında ve her durum tahmini için bu doğrusallaştırmayı tekrarlar (yeniden doğrusallaştırma). Bu sürekli yeniden doğrusallaştırma, özellikle sistemdeki doğrusal olmayanlıkların belirgin olduğu durumlarda tahmin doğruluğunu artırabilir [39]. Ayrıca, faktör grafikleri, ölçümlerin toplu (batch) olarak işlenmesi prensibine dayanır [27]. Bu yaklaşım, geçmiş ve mevcut zaman adımları arasındaki zamansal korelasyonları daha etkin bir şekilde kullanmasını sağlar. Pratikte bu, her ölçüm güncellemesinden sonra hem bir ileri yönlü filtreleme hem de bir geri yönlü düzeltme (smoothing) işlemi yapılmasına eşdeğerdir [27].

Toplu işlemenin potansiyel bir dezavantajı olan hesaplama yükü ve gerçek

zamanlılıktan ödün verme endişesi, faktör grafiklerinin yapısal özellikleriyle giderilebilmektedir. Grafiğin genellikle seyrek olması, yani her durumun yalnızca az sayıda diğer durum ve ölçümle ilişkili olması, optimizasyon sürecinin verimli bir şekilde yürütülmesini sağlar [27]. Hesaplama maliyetini daha da azaltmak ve gerçek zamanlı performansı korumak için, tüm geçmiş yerine yalnızca belirli bir zaman aralığını kapsayan durumların optimize edildiği kayan pencere (sliding window) veya sabit gecikmeli düzeltme (fixed-lag smoothing) benzeri yaklaşımlar tercih edilmektedir. Daha gelişmiş bir yöntem olan Artımlı Düzeltme ve Haritalama (Incremental Smoothing and Mapping, iSAM2) ise, yeni ölçümler eklendikçe tüm grafiği yeniden optimize etmek yerine, faktör grafiğini verimli bir şekilde temsil eden bir Bayes ağacına dönüştürerek ve yalnızca etkilenen küçük bir bölümü güncelleyerek gerçek zamanlı performansı mümkün kılar [40].

Bu avantajlarla, faktör grafiği optimizasyonunun sunduğu iteratif iyileştirme, gelişmiş nonlinear yönetimi ve sensör füzyonundaki genel gücü ile bu tez çalışmasındaki VIO algoritmasının temelini oluşturmaktadır.

3.4.4 Ataletsel Sensör Ön Entegrasyonu

VIO algoritmalarında 6DOF pose kestirimi için IMU, yüksek frekansta (örn. 200/1000 Hz) ivme ve açısal hız bilgisi üretir. Ancak, sistemin global konumunu düzelten kameralar, GNSS alıcıları veya lazer tarayıcılar gibi diğer sensörler genellikle çok daha düşük frekanslarda (20-30 FPS) veri üretirler. Bu sensörlerin farklı veri hızları, durum kestirimi için yaygın olarak kullanılan faktör grafı tabanlı optimizasyon yaklaşımlarında önemli bir zorluğa yol açar.

Faktör grafı tabanlı optimizasyon çerçevelerinde, IMU verisinin yüksek frekansına karşılık her bir IMU okuması için ayrı bir gizli değişken düğümünün ve dinamik faktörün eklenmesi, iki harici ölçüm arasındaki aralıkta aşırı sayıda düğümün zincirlenmesine yol açar. Bu "düğüm patlaması" (node proliferation) durumu, grafın boyutu ve yoğunluğunu artırarak optimizasyon algoritmalarının hesaplama yükünü ve süresini katlanarak yükseltir. Bu durum sistemin gerçek zamanlı çalışmasını veya büyük ölçekli ortamlarda ölçeklenebilirliğini ciddi şekilde kısıtlar. Bu karmaşık ve hesaplama açısından pahalı problemi çözmek amacıyla, literatürde "ön entegrasyon" (pre-integration) ve "akıllı faktörler" (smart factors) yaklaşımları geliştirilmiştir.

Ön entegrasyon işlemi, sadece ataletsel ölçümlerle birbirine zincirlenmiş birden fazla düğümün, birden fazla ataletsel güncellemeyi tek bir faktör düğümünde temsil ederek gizli değişken düğüm sayısında önemli bir azalma sağlar [41, 42]. Graf daha

seyrek hale gelir; bu durum optimizasyon sürecinin hesaplama maliyetini düşürür ve hesaplama hızını artırır, gerçek zamanlı uygulamalar için daha uygun hale getirir.

Akıllı faktörler (Smart Factors), Carlone vd. (2014) tarafından önerilen, faktör grafındaki gizli değişkenleri Hedef Değişkenler (Target Variables) ve Destek Değişkenler (Support Variables) olmak üzere iki ana sınıfa ayırarak hesaplama verimliliğini artıran bir yaklaşımdır [43]. Hedef değişkenler (Target Variables), sistemin doğrudan kestirmeyi amaçladığı Anahtar Durum (Keyframe) pozlarını temsil ederken; Destek değişkenler (Support Variables), hedef değişkenlerin daha doğru ve tutarlı bir şekilde kestirilmesini sağlamak amacıyla geçici olarak veya ara basamak olarak kullanılan değişkenlerdir. Ön entegrasyonun (Pre-integration) uygulanmadığı veya kısmi uygulandığı durumlarda, iki anahtar kare (hedef değişken) arasındaki her bir yüksek frekanslı IMU okumasına karşılık gelen ara sistem durumları (pozisyon, hız, yönelim), örneğin bir saniye içinde 100 IMU okuması varsa, bu 100 ara IMU durumu birer destek değişkeni olabilir ve yine kısa süreliğine görünen ve hemen kaybolan, kalıcı bir harita noktası olarak kullanılmayacak olan görsel özelliklerin 3D konumları destek değişkenleridir.

Eğer bir grup destek değişkeni yalnızca kendi aralarında ve/veya bazı hedef değişkenlerle sınırlı bir etkileşime sahipse ve doğrudan tüm hedef değişkenlerle etkileşim içinde değilse, bu destek değişkenleri tek bir düğümde birleştirilebilir (marginalize edilebilir).

Bir kez destek değişkenlerinin marjinalleştirilmesi, destek değişkenlerin bireysel değerlerini ve ilişkili ayrıntılı belirsizlik bilgilerinin kaybolmasına neden olmaktadır. Eğer gelecekte, bu marjinalleştirilmiş değişkenlerin etkileşimde olduğu hedef değişkenlerin (örneğin, robotun o anki pozu) değerleri tahmininizden önemli ölçüde saparsa veya çok büyük bir değişiklik yaşarsa, marjinalleştirme işlemi sırasında yapılan doğrusallaştırma varsayımları geçerliliğini yitirir. Bu durumda, en iyi çözümü elde etmek için marjinalleştirme işleminin geri alınması düşünülebilir ancak tüm değişkenleri baştan optimize etmeniz gerektiğinden bu durum imkansız varsayılabilir ve marjinalleşmiş bilgiye dayalı olarak yeni, hatalı bir optimizasyon yapma riskiyle karşılaşılır [44].

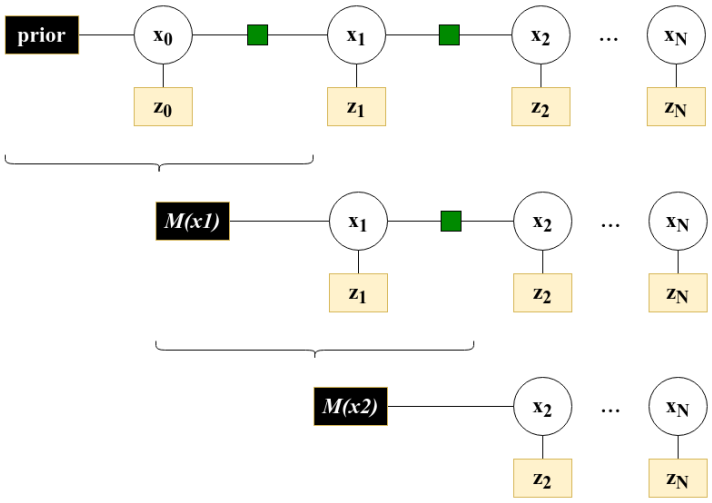
Bu durumlar, marjinalleştirme işleminin dikkatli bir şekilde verilmesini gerektirir. Genellikle, çok eski ve sistemin güncel durumu için daha az kritik olan değişkenler marjinalleştirilir. Eğer bir değişkenin gelecekteki değerlerinin veya ilişkilerinin çok dinamik ve öngörülemez bir şekilde değişmesi bekleniyorsa, o değişkeni marjinalleştirmek riskli olabilir. Bu, optimizasyonun esnekliği ile hesaplama verimliliği arasında bir denge kurulmasını zorunlu kılar [44].

3.4.5 Durum Tahmini Optimizasyonunda Marjinalleştirme ve Kayan Pencereler

Faktör grafiklerindeki büyük boyutlu optimizasyon problemleri ile başa çıkmak ve hesaplama maliyetini yönetmek için marjinalleştirme (marginalize) ve kayan pencereler (sliding window) stratejileri kritik öneme sahiptir.

Marjinalleştirme, faktör grafiğindeki belirli gizli değişkenlerin (hidden variables) sistemden deterministik veya olasılıksal olarak çıkarılması işlemidir. Bu işlem, genellikle bu değişkenlere ait bilgiyi, geride kalan (bağlantılı) değişkenler üzerinde yeni bir faktör (veya faktörler) olarak yoğunlaştırarak yapılır. Temel amaç, grafikteki düğüm sayısını azaltmak ve böylece optimizasyon probleminin boyutunu düşürmektir. Bu, özellikle sınırlı hesaplama kaynaklarına sahip platformlarda veya uzun süreli operasyonlarda gerçek zamanlı performansı sürdürmek için elzemdir. Grafiksel olarak bu işlem, elimine edilen düğümün grafikten çıkarılması ve onun komşuları arasında ortaya çıkan tüm yeni bağımlılıkların korunması anlamına gelir.

Marjinalleştirme, özellikle kayan pencereler yaklaşımı ile birlikte kullanıldığında güçlü bir optimizasyon mekanizması sunar. Kayan pencere stratejisinde, optimizasyon sadece belirli boyuttaki pencere içerisindeki durum ve gözlemlerin (genellikle eskimiş olan) birleştirilmesini kapsar.



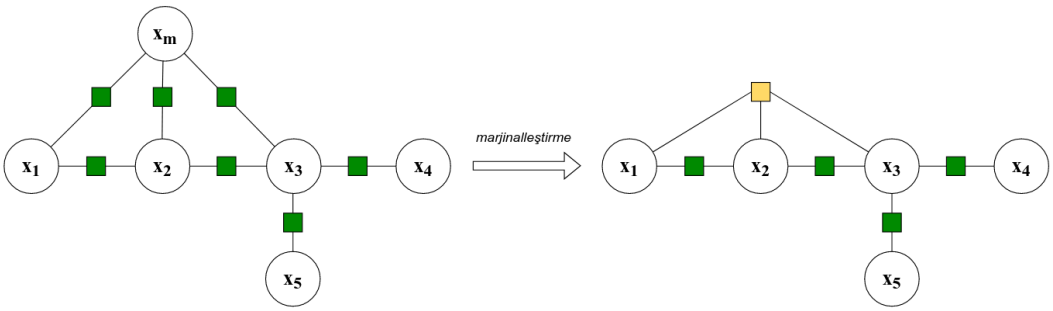
Şekil 3.5 Marginalize işleminin faktör graf uygulanması

Şekil 3.5, bu mekanizmayı şematik olarak göstermektedir. Başlangıçta bir *prior* faktörü ile x_0 durumu, x_1, x_2 gibi sonraki durumlar ve bunlarla ilişkili z_0, z_1, z_2 ölçüm faktörleri bulunmaktadır. İlk marjinalleştirme adımında, en eski durum olan x_0 grafikten çıkarılır. Bu çıkarılan durumun bilgisi, yeni bir $M(x_1)$ faktörü olarak x_1 durumuna aktarılır. Bu $M(x_1)$ faktörü, x_0'dan gelen tüm birikmiş bilgiyi

özetleyen bir köprü görevi görür. Bir sonraki adımda, pencere ileri kayar, x_1 marjinalleştirilir ve bilgisi $M(x_2)$ olarak x_2'ye aktarılır. Bu döngü, pencerenin boyutunu sabit tutarak hesaplama karmaşıklığının sabit bir üst sınır içinde kalmasını sağlar [45].

3.4.5.1 Fill-in Fenomeni

Seyrek doğrusal sistemlerin çözümünde kullanılan Cholesky ayrıştırması gibi doğrudan çözüm yöntemlerinde, matrisin yapısındaki seyrekliğin korunması hesaplama verimliliği açısından kritik öneme sahiptir. Ancak marjinalleştirme işlemleri sırasında başlangıçta sıfır olan bazı matris elemanları sıfır olmayan değerlere dönüşebilir. Bu durum fill-in (doldurma) olarak adlandırılmaktadır [44].



Şekil 3.6 x_m değişken düğümünün marjinalleştirilmesi

Şekil 3.6'de fill-in oluşumuna dair grafiksel bir örnek sunulmuştur. Sol tarafta verilen Markov Rastgele Alanı (MRF), bilgi matrisi Şekil 3.36'da Λ ile temsil edilirken, sağ tarafta elimine edilen değişkenlere bağlı olarak elde edilen Bayes ağı ve buna karşılık gelen Cholesky faktörü, Şekil 3.37'de R gösterilmektedir.

Bir Gauss Markov Rastgele Alanı (MRF) için, değişkenler arasındaki koşullu bağımlılıklar duyarlılık matrisi (Λ) ile temsil edilir. Λ_{ij} elemanı, x_i ve x_j değişkenleri arasında doğrudan bir koşullu bağımlılık varsa sıfırdan farklıdır. Diyagonal elemanlar (Λ_{ii}) ise daima sıfırdan farklıdır. Şekil 3.6'de gösterilen faktör grafiği için, değişkenleri $x_1, x_2, x_3, x_4, x_5, x_m$ şeklinde sıralayarak oluşturulan başlangıçtaki Λ matrisinin seyrek yapısı Şekil 3.36 ile ifade edilmektedir:

$$\Lambda = \begin{pmatrix} \Lambda_{11} & \Lambda_{12} & & & & \Lambda_{1m} \\ \Lambda_{21} & \Lambda_{22} & \Lambda_{23} & & & \Lambda_{2m} \\ & \Lambda_{32} & \Lambda_{33} & \Lambda_{34} & \Lambda_{35} & \Lambda_{3m} \\ & & \Lambda_{43} & \Lambda_{44} & & \\ & & \Lambda_{53} & & \Lambda_{55} & \\ \Lambda_{m1} & \Lambda_{m2} & \Lambda_{m3} & & & \Lambda_{mm} \end{pmatrix} \tag{3.36}$$

Matris içinde boş bırakılan yerler, ilgili değişkenler arasında başlangıçta doğrudan bir koşullu bağımlılık olmadığını (yani Λ_{ij}'nin sıfır olduğunu) gösterir. Özellikle Λ_{13} ve Λ_{31} elemanları, x_1 ve x_3 değişkenleri arasında doğrudan bir bağlantı bulunmadığını belirtmektedir.

x_m değişkeni marjinalize edildikten sonra, kalan değişkenler $(x_1, x_2, x_3, x_4, x_5)$ için yeni bir duyarlılık matrisi (Λ') oluşur. Bu Λ' matrisi, orijinal Λ matrisinin Schur tamamlayıcısıdır. Bu yeni duyarlılık matrisinin Cholesky ayrıştırması $(\Lambda' = R^T R)$ yapılarak üst üçgen R faktörü elde edilir. Kalan değişkenlerin sıralaması x_1, x_2, x_3, x_4, x_5 olarak kabul edildiğinde, R matrisinin yapısı ve doldurma elemanları Şekil 3.37 ile ifade edilmiştir.

$$R = \begin{pmatrix} R_1 & T_{12} & T_{13} & & \\ & R_2 & T_{23} & & \\ & & R_3 & T_{34} & T_{35} \\ & & & R_4 & \\ & & & & R_5 \end{pmatrix} \tag{3.37}$$

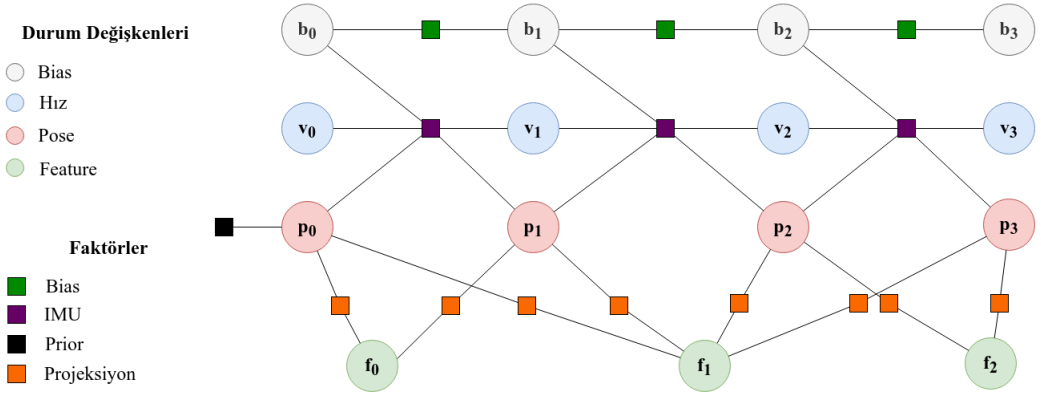
Eleme sonrası elde edilen üst üçgen matris R'nin elemanları, kalan değişkenler arasındaki güncellenmiş koşullu bağımlılık yapısını yansıtmaktadır. Köşegen elemanları (R_i), ilgili x_i değişkeninin, koşullu kesinliğini (conditional precision) temsil eden sıfırdan farklı değerlerdir. Köşegen dışı elemanlar (T_{ij}) ise koşullu bağımlılıkları ifade eder. Bunlardan $T_{12}, T_{23}, T_{34}, T_{35}$ gibi elemanlar, orijinal faktör grafiğinde zaten mevcut olan doğrudan bağımlılıklara karşılık gelir ve eliminasyon sonrasında da sıfırdan farklı kalır. Ancak, T_{13} gibi kırmızı ile belirtilen elemanlar fill-in fenomenini yansıtır. Bu eleman, başlangıçtaki Λ matrisinde ilgili eleman (Λ_{13}) sıfır olmasına rağmen, x_m değişkeninin eliminasyonu nedeniyle x_1 ve x_3 arasında indüklenen yeni bir koşullu bağımlılığı ifade eder.

Fill-in, doğrudan değişkenlerin eliminasyon sırasına bağlı olarak ortaya çıkar ve faktör matrislerinin seyrekliğini olumsuz etkileyerek bellek kullanımı ile hesaplama süresini önemli ölçüde artırabilir. Bu nedenle, uygun bir eliminasyon sıralaması belirlemek fill-in oluşumunu minimize etmek açısından oldukça önemlidir. Genellikle, minimum derece bazlı sıralama algoritmaları, özellikle COLAMD (Column Approximate Minimum Degree) algoritması, etkin bir sıralama yaklaşımıdır [46]. Amacı, adım adım ilerleyerek, her bir eliminasyon adımında ortaya çıkan fill-in sayısını yaklaşık olarak en aza indirmektir. Bu, nihai Cholesky

faktörü R'nin daha fazla sıfır eleman içermesini sağlar.

3.5 VINS-Fusion

VIO, kamera ve IMU sensörlerinden gelen verileri sıkı-bağlı (tightly-coupled) bir şekilde birleştirerek bir anın göreceli pozunu ve yönelimini yüksek frekansta ve hassasiyetle tahmin etme yeteneğine sahiptir. Tong Qin ve çalışma arkadaşları tarafından sunulan VINS-Mono [10], sağlam bir başlatma, çevrimiçi dışsal kalibrasyon ve 4 serbestlik dereceli (4-DoF) poz grafiği optimizasyonu içeren bir döngü kapatma modülü sunmaktadır. VINS-Fusion ise VINS-Mono'nun bir uzantısı ve çoklu sensör desteği (stereo, mono+IMU, stereo+IMU ve GNSS desteği) sunan çalışmadır [11, 47, 48].



Şekil 3.7 Görsel-Ataletsel Odometri için temsili faktör grafiği

Şekil 3.7, VIO faktör grafiğinin temsili örneğini sunmaktadır. Bu graf yapısı, VIO problemini bir optimizasyon problemi olarak formüle eder. Dairesel düğümler, sistemin optimize edilmeye çalışılan durum değişkenlerini; farklı zamanlardaki poz (p), hız (v), IMU sapması (b) ve gözlemlenen özniteliklerin (f) 3B konumları—temsil eder. Karesel düğümlerle gösterilen faktörler ise bu değişkenler arasındaki matematiksel kısıtları ifade eder. Ancak, bu yapıdaki tüm kısıtlar (IMU ve projeksiyon) göreceli ölçümlere dayandığı için, zamanla biriken sürüklenme hatasını tek başına engelleyemez. Özellikle uzun mesafeli veya uzun süreli görevlerde sınırsızca büyüyebilir ve bu durum zamanla biriken sürüklenme (drift) hatasına neden olur. Sürüklenmeyi ortadan kaldırmak ve sistemi global bir referans çerçevesine oturtmak için mutlak konum bilgisinin sistemi beslemesi kritik öneme sahiptir. Eşleme tabanlı AVL teknikleri, VIO sistemlerinin karşılaştığı bu temel soruna çözüm sunma potansiyeli taşımaktadır. Bu tez çalışması, VIO sistemlerindeki sürüklenme birikimini, önceden oluşturulmuş bir harita üzerinde anlık görüntüleri eşleyerek mutlak konum tespiti yapan AVL yöntemleri ile düzeltme olanaklarına odaklanmaktadır.

Bu hedefe yönelik olarak, AVL tekniğinden elde edilecek mutlak konum verisinin popüler bir VIO çatısı olan VINS-Fusion'a nasıl entegre edilebileceğini anlamak kritik bir adımdır. Bu bölümde, VINS-Fusion algoritmasının, temel VINS-Mono mimarisini genişleterek mevcut bir mutlak konum kaynağı olan GNSS verisini global optimizasyon (pose graph optimization) adımına nasıl dahil ettiği referans bir model olarak incelenecektir. Bu inceleme, AVL'den türetilen görsel konumlandırma verisinin de benzer bir mantıkla sisteme beslenerek sürüklenme hatasının etkin bir şekilde düzeltilmesi için bir yol haritası sunacaktır.

3.5.1 VINS Global Optimizasyon

VINS-Mono, anlık durum kestirimi sürecinde kayan pencere optimizasyonu (sliding window optimization) yaklaşımını benimsemektedir [10]. Bu yöntem, hesaplama yükünü kontrol altında tutmak amacıyla yalnızca en güncel ve sınırlı sayıdaki sensör ölçümünü içeren dinamik bir pencere üzerinde optimizasyon gerçekleştirir. Bu sayede, yüksek frekansta konum tahmini yapılması ve kısa vadeli tutarlılığın sağlanması mümkün olmaktadır. Ancak, bu lokal optimizasyon yapısı zaman içerisinde biriken kümülatif sürüklenme hatasına (accumulative drift error) karşı tek başına yeterli değildir. Uzun süreli görevlerde oluşan bu hata, sistemin global tutarlılığını olumsuz etkileyebilmektedir.

Bu sorunu aşmak amacıyla VINS-Mono, döngü kapatma (loop closure) mekanizmasını temel alan bir global optimizasyon stratejisi uygulamaktadır. Söz konusu strateji, harita üzerinde geçmiş pozisyonlarla yeni gözlemler arasındaki benzerliklerin tespit edilmesiyle başlamakta; ardından, bu benzerlikler aracılığıyla sistemin küresel tutarlılığı yeniden yapılandırılmaktadır. Bu süreç, özellikle sistemin yeniden konumlandırma (relocalization) aşamasında kritik rol oynamakta ve biriken hataların giderilmesinde belirleyici olmaktadır.

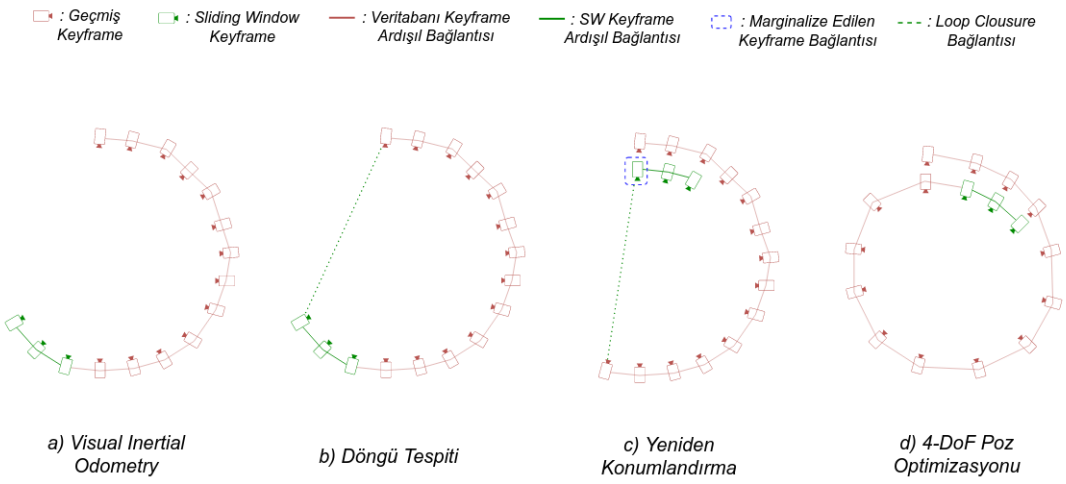
Global optimizasyon süreci, temel olarak özellik çıkarımı ve döngü kapatma tespiti ile başlamaktadır. Bu aşamada sistem, sahne yeniden tanıma işlemi için `DBoW2` kütüphanesinden yararlanır. Mevcut anahtar kareden elde edilen BRIEF tanımlayıcıları (descriptors), daha önce oluşturulmuş bir kelime çantası (Bag-of-Words) veritabanı üzerinde sorgulanır. Bu veritabanı, geçmiş tüm anahtar karelere ait tanımlayıcıları içermektedir. Potansiyel döngü kapatma adayları ise zamansal tutarlılık (temporal consistency) da dikkate alınarak Bag-of-Words sorgusu sonucunda elde edilen benzerlik skorlarına göre belirlenir [10].

Olası bir döngü adayı tanımlandıktan sonra, yalnızca görsel benzerliğe dayalı karar vermek yeterli olmadığından, sistem geometrik doğrulama adımına geçer. Bu

aşamada, mevcut anahtar karedeki 2B öznitelikler ile döngü adayı olan geçmiş anahtar kareye ait 3B harita noktaları eşleştirilir. Elde edilen 2B-3B eşleşmeleri, bir RANSAC (Random Sample Consensus) çerçevesi içinde değerlendirilerek, PnP (Perspective-n-Point) algoritması yardımıyla mevcut kameranın pozisyonu ve yönelimi hesaplanır. Eğer yeterli sayıda inlier eşleşme bulunursa, döngü geometrik olarak doğrulanmış kabul edilir ve hesaplanan poz, bir `SE(3)` (Special Euclidean Group in 3 dimensions) dönüşümü olarak ifade edilir [47].

Ancak, monoküler VIO sistemlerinin doğasında var olan ölçek belirsizliği (scale ambiguity) nedeniyle, döngü kapatma kısıtı doğrudan `SE(3)` yerine, ölçek bilgisini de içeren bir `Sim(3)` (Similarity transformation in 3 dimensions) dönüşümü ile modellenir. Bu `Sim(3)` kısıtı; yönelim, konum ve ölçek parametrelerini kapsayacak şekilde tanımlanır. Hesaplanan bu dönüşüm, sistemin poz grafı üzerinde, mevcut anahtar kare düğümü ile geçmişteki eşleşen anahtar kare düğümü arasına, yüksek güvenilirlikli bir döngü kapatma kenarı (loop closure edge) olarak eklenir.

Bu yeni ve güçlü global kısıtın graf yapısına dâhil edilmesi, sistemde bir Poz Grafiği Optimizasyonu (Pose Graph Optimization, PGO) sürecini tetikler. PGO, döngü kapatma kısıtının sağladığı bilgiyi kullanarak, graf üzerindeki tüm anahtar kare pozisyonlarını küresel olarak yeniden hizalar. Bu optimizasyon işlemi, zaman içerisinde birikmiş olan sürüklenme hatasının tüm yörüngeye dengeli bir şekilde dağıtılmasını sağlar. Böylece sistem, uzun vadeli küresel doğruluk ve tutarlılığı yeniden kazanmış olur.



Şekil 3.8 VIO döngü kapatma ve poz grafiği optimizasyon gösterimi

Bu süreç Şekil 3.8'de şematik olarak gösterilmektedir. Şekil 3.8(a)'da, Görsel-Ataletsel Odometri' nin sensör hataları nedeniyle zamanla biriken

sürüklenme hatası (drift) görülmektedir. Şekil 3.8(b) ise, mevcut bir anahtar kare (yeşil) ile geçmişteki bir anahtar kare (kırmızı) arasında potansiyel bir döngünün tespit edildiği anı ve bu iki anahtar kare arasında kurulan geometrik kısıtı yeşil noktalı çizgi ile göstermektedir. Bu tespitin ardından, Şekil 3.8(c)'de, ilk etapta yalnızca kayan pencere içerisindeki pozların yerel olarak düzeltildiği bir yeniden konumlandırma işlemi gerçekleşebilir; burada mavi kesikli çerçeve ile işaretlenmiş anahtar kare, marjinalize edilen bir anahtar kareyi temsil etmektedir. Nihayetinde, Şekil 3.8(d)'de, tespit edilen döngü kapatma kısıtının kullanılarak tüm yörüngedeki hatanın dağıtıldığı ve global olarak tutarlı bir poz grafiğinin elde edildiği 4 serbestlik dereceli (4-DoF) poz optimizasyonu sonucu gösterilmektedir. Bu optimizasyon, tüm anahtar karelerin pozlarını düzelterek genel harita tutarlılığını artırır.

3.5.1.1 Poz Grafiği

Global optimizasyon süreci, sistem tarafından belirli aralıklarla oluşturulan anahtar karelerin (keyframes) bir poz grafiği üzerinde yürütülür. Bu graf yapısı, temel olarak düğümler ve kenarlardan oluşur. Düğümler (Nodes), her bir anahtar karenin dünya (W) referans çerçevesindeki pozunu ($\mathbf{p}_w^{b_k}$ pozisyonu ve $\mathbf{q}_w^{b_k}$ ise yönelimi) $\mathbf{x}_k = \{\mathbf{p}_w^{b_k}, \mathbf{q}_w^{b_k}\}$ olarak temsil eder.

Kenarlar (Edges) ise bu düğümler arasındaki kısıtları tanımlar ve iki temel türde bulunur. Birincisi, ardışık kenarlar (sequential edges) olup, VIO ön-ucundan (frontend) gelen ve ardışık anahtar kareler (k ve $k+1$) arasındaki göreceli hareket bilgilerini içeren kısıtlardır (Şekil 3.8'deki kırmızı ve yeşil düz çizgiler). İkincisi ise döngü kapatma kenarlarıdır (loop closure edges). Bu kenarlar, mevcut bir anahtar kare i ile geçmişteki benzer bir anahtar kare j arasında, `DBoW2` kullanılarak saptanan görsel benzerlik ve geometrik tutarlılık doğrulandığında poz grafiğine eklenen göreceli poz kısıtlarıdır (Şekil 3.8(c)'deki yeşil kesikli çizgi).

3.5.1.2 4-DoF Optimizasyonu

VINS-Mono algoritmasının öne çıkan yeniliklerinden biri, global poz grafiği optimizasyonunu geleneksel 6 serbestlik derecesi (6-DoF) yerine, 4 serbestlik derecesi (4-DoF) üzerinden gerçekleştirmesidir. Bu yaklaşımın temel dayanağı, ataletsel çlçüm biriminin ivmeölçer bileşeninin yerçekimi vektörünü yüksek hassasiyetle ölçebilmesidir. Bu sayede, sistemin yatış (ϕ) ve dikilme (θ) açıları global olarak gözlemlenebilir hale gelir ve bu iki serbestlik derecesindeki zamanla biriken drift ihmal edilebilir düzeyde kalır. Sonuç olarak, kümülatif hata esasen üç boyutlu pozisyon ($\mathbf{p}^w = [x, y, z]^T$) ve sapma (yaw, ψ) açısında yoğunlaşır [48].

Bu doğrultuda optimizasyon, yalnızca bu belirtilen 4 serbestlik derecesi üzerinden yürütülür. Bir döngü kapatma kenarı ile birbirine bağlanan i ve j anahtar kareleri arasındaki hata (residual) vektörü $\mathbf{r}_{ij}$, Denklem 3.38'te gösterildiği gibi tanımlanır:

$$\mathbf{r}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \begin{bmatrix} \mathbf{R}(\phi_i, \theta_i)^T(\mathbf{p}_j^w - \mathbf{p}_i^w) - \hat{\mathbf{p}}_{ij}^b \\ (\psi_j - \psi_i) - \hat{\psi}_{ij} \end{bmatrix} \tag{3.38}$$

Denklem 3.38 de $\mathbf{x} = [\mathbf{p}^w, \psi]^T$ optimize edilecek durum vektörünü; $\mathbf{p}_i^w$ ve $\mathbf{p}_j^w$, sırasıyla i ve j anahtar karelerinin optimize edilecek global pozisyon vektörlerini; ψ_i ve ψ_j ise bu karelerin optimize edilecek global sapma açılarını temsil eder. $\hat{\mathbf{p}}_{ij}^b$ ve $\hat{\psi}_{ij}$ terimleri, i ve j pozları arasında döngü kapatma ile ölçülen göreceli pozisyonu ve göreceli sapma açısını ifade eder; burada üst simge b, ölçümün gövde (body) koordinat sisteminde yapıldığını belirtir. $\mathbf{R}(\phi_i, \theta_i)$ ise, i anında IMU ile global olarak bilinen yatış (ϕ_i) ve dikilme (θ_i) açılarından elde edilen rotasyon matrisidir [10].

Sıralı kenarların ve döngü kapatma kenarlarının oluşturduğu poz grafiği, aşağıdaki maliyet fonksiyonunun minimize edilmesiyle optimize edilir:

$$\min_{\mathbf{p},\psi} \left\{ \sum_{(i,j)\in\mathcal{S}} \|\mathbf{r}_{i,j}\|^2 + \sum_{(i,j)\in\mathcal{L}} \rho\left(\|\mathbf{r}_{i,j}\|^2\right) \right\} \tag{3.39}$$

Burada $\mathcal{S}$ tüm sıralı kenarlar kümesini ve $\mathcal{L}$ ise döngü kapatma kenarları kümesini belirtir. Sıkı-bağlaşımlı yeniden yerelleştirme (tightly-coupled relocalization) yanlış döngü kapatmalarını zaten elemeye yardımcı olsa da, olası yanlış döngülerin etkisini daha da azaltmak için döngü kapatma kenarlarına ek bir Huber normu $\rho(\cdot)$ uygulanır. Optimizasyon, tüm pozların 4-DoF durumları ($\mathbf{p}^w, \psi$) üzerinden gerçekleştirilir. Bu süreç, poz grafiğindeki tutarsızlıkları gidererek birikmiş sürüklenmeyi grafın tamamına dağıtır ve global olarak daha tutarlı bir harita ve pozisyon kestirimi sağlar.

3.5.2 GNSS Verisinin Global Optimizasyon Grafiğine Entegrasyonu

VINS-Fusion, Şekil 3.7 ile açıklanan poz grafiği (pose graph) yapısını temel almakta olup, bu yapıya ek olarak GNSS faktörünü de entegre etmektedir.

3.5.2.1 Koordinat Çerçeveleri ve Hizalama Problemi

VINS-Fusion sisteminde GNSS verilerinin Görsel-Ataletsel Odometri ile bütünleştirilmesinde karşılaşılan temel zorluk, farklı koordinat sistemlerinin eş zamanlı olarak kullanılmasıdır. Bu kapsamda iki ana referans çerçevesi dikkate alınmaktadır:

- **VIO Dünya Çerçevesi (W):** Sistemin başlatıldığı anda tanımlanan ve orijini o noktaya yerleştirilen keyfi bir kartezyen koordinat sistemidir.
- **GNSS Çerçevesi:** GNSS verileri, genellikle WGS84 standardına uygun olarak, enlem, boylam ve yükseklik (Latitude, Longitude, Altitude, LLA) formatında sağlanmaktadır.

VIO' nun GNSS bilgiisi ile mutlak konum düzeltmesi alabilmesi için, VIO çerçevesi ile uyumlu olacak şekilde yerel kartezyen koordinat sistemine dönüştürülmesi gerekmektedir. VINS-Fusion, bu amaçla, ilk GNSS ölçümünü orijin olarak kabul eden bir Doğu-Kuzey-Yukarı (ENU) çerçevesi kullanmaktadır.

Bu işlem sonrasında, çözülmesi gereken problem, VIO sistemine ait keyfi W çerçevesi ile GNSS ölçümlerine dayalı yerel ENU çerçevesi arasındaki üç boyutlu benzeşim dönüşümünün (3D similarity transformation) belirlenmesidir. Söz konusu dönüşüm, bir dönme matrisi $\mathbf{R}_w^{enu}$ ile bir öteleme vektörü $\mathbf{p}_w^{enu}$'dan oluşmaktadır.

3.5.2.2 GNSS Faktör İmplementasyonu

Bir k anahtar karesi için hem VIO pozisyonu hem de senkronize bir GNSS ölçümü mevcut olduğunda, poz grafiğine yeni bir tekli faktör (unary factor) eklenir. Bu faktör, VIO tarafından tahmin edilen konum ile GNSS ölçümünün ne derece uyumlu olduğunu değerlendiren bir hata fonksiyonu tanımlar.

Bu bağlamda, ilgili dönüşüm ve hata modellemesi yapılırken çeşitli büyüklükler tanımlanır. $\mathbf{p}_{b_k}^w$ değişkeni, k zamanında VIO tarafından tahmin edilen gövde (body) çerçevesinin dünya koordinat sistemindeki (W) konumunu temsil eder. GNSS tarafından aynı zaman anında ölçülen anten pozisyonu $\mathbf{p}_{gnss}^{enu}$ ile ifade edilir ve bu pozisyon, ENU koordinat sistemine göre tanımlanmıştır. GNSS anteninin gövde çerçevesine göre sabit ötelemesi ise $\mathbf{p}_b^{gnss}$ ile gösterilir ve bu değer sistem öncesinde yapılan kalibrasyon işlemleriyle belirlenir. VIO'nun referans aldığı dünya koordinat sisteminin ENU sistemine göre konumlandırılması, $\mathbf{R}_w^{enu}$ (döndürme matrisi) ve $\mathbf{p}_w^{enu}$ (öteleme vektörü) ile gerçekleştirilir. VIO tarafından tahmin edilen gövde

konumunun GNSS ölçüm sistemine taşınması için aşağıdaki dönüşüm ifadesi kullanılır:

$$\hat{\mathbf{p}}_{gnss}^{enu} = \mathbf{R}_{w}^{enu}(\mathbf{p}_{b_k}^{w} + \mathbf{R}_{b_k}^{w}\mathbf{p}_{b}^{gnss}) + \mathbf{p}_{w}^{enu} \tag{3.40}$$

Bu denklemde $\mathbf{R}_{b_k}^{w}$, VIO tarafından k anındaki anahtar kare için elde edilen gövde yönelimidir. Elde edilen $\hat{\mathbf{p}}_{gnss}^{enu}$ vektörü, GNSS anteninin VIO üzerinden tahmin edilen ENU koordinat sistemindeki pozisyonunu ifade eder. GNSS faktörü ile ilişkili hata vektörü $\mathbf{r}_{gnss,k}$ ise, tahmin edilen bu pozisyon ile gerçek GNSS ölçümü arasındaki farktan elde edilir ve aşağıdaki şekilde tanımlanır:

$$\mathbf{r}_{gnss,k} = \hat{\mathbf{p}}_{gnss}^{enu} - \mathbf{p}_{gnss}^{enu} \tag{3.41}$$

Bu hata terimi, global optimizasyon problemine ilave edilerek güncellenmiş toplam maliyet fonksiyonu şu şekilde ifade edilir:

$$\min_{\mathcal{X},\mathbf{R}_{w}^{enu},\mathbf{p}_{w}^{enu}} \left(\sum_{\text{seq}} \|\mathbf{r}_{\text{seq}}\|^2 + \sum_{\text{loop}} \rho\left(\|\mathbf{r}_{\text{loop}}\|_{\Sigma_{loop}}^2 \right) + \sum_{\text{gnss}} \rho\left(\|\mathbf{r}_{\text{gnss,k}}\|_{\Sigma_{gnss}}^2 \right) \right) \tag{3.42}$$

GNSS ölçüm hatalarındaki aykırı (outlier) değerlerin optimizasyon üzerindeki etkisini azaltmak amacıyla, GNSS farkına Huber normu $\rho(\cdot)$ uygulanır. Ayrıca, Σ_{gnss} terimi, GNSS ölçümüne ait kovaryans matrisini temsil eder ve hata teriminin güvenilirlik derecesini yansıtır.

Döngü kapatma sürecini de kapsayan GNSS bilgisinin VIO sistemine implementasyon adımı aşağıdaki işlem adımları ile ifade edilebilir:

- **Koordinat Dönüşümü:** İlk olarak GNSS LLA verisi, yerel ENU çerçevesine dönüştürmek için kullanılır.
- **Hizalama Başlatma:** Sistem, VIO ve ENU çerçeveleri arasındaki $(\mathbf{R}_{w}^{enu}, \mathbf{p}_{w}^{enu})$ hizalamasını hemen hesaplamaz. Sistemin güvenilir bir hizalama tahmini için yeterli sayıda GNSS ölçümü ve aracın yeterli bir mesafe kat etmesini beklediği görülür. Bu, hizalamanın gözlemlenebilirliğini artırır.
- **Faktör Ekleme:** Hizalama başarılı bir şekilde başlatıldıktan sonra, gelen

her yeni GNSS ölçümü için ilgili fonksiyon, Ceres Solver optimizasyon problemine özel bir maliyet fonksiyonu (GNSS Factor) ekler. Bu faktör, Denklem 3.41'de formüle edilen hatayı uygular.

- **Optimizasyon:** Döngü kapatma veya yeni bir GNSS faktörü eklendiğinde, Ceres Solver kullanarak hem anahtar karelerin 4-DoF pozlarını hem de GNSS hizalama parametrelerini içeren Denklem 3.42'deki genişletilmiş maliyet fonksiyonu minimize edilir.

Bu entegrasyon sayesinde, poz grafiği artık sadece kendi içinde tutarlı olmakla kalmaz, aynı zamanda GNSS tarafından sağlanan mutlak ölçümlerle küresel olarak çivilenmiş (anchored) olur. Bu, VIO'nun en büyük zayıflığı olan konum ve sapma sürüklenmesini etkili bir şekilde ortadan kaldırır.

4
YÖNTEM

4.1 AVL ve VINS-Fusion Tabanlı Navigasyon Sistemi

Bu bölümde, Görsel-Ataletsel Odometri sistemlerinde zamanla biriken pozisyon sürüklenmesi sorununa çözüm geliştirmek amacıyla, yalnızca iki serbestlik derecesine (enlem ve boylam) sahip Mutlak Görsel Konumlandırma çıktılarının, VINS-Fusion algoritmasına harici bir küresel konumlandırma kaynağı olarak entegrasyonu teknik bir bakış açısıyla ele alınacaktır. Özellikle, AVL tarafından sağlanan enlem ve boylam verilerinin VINS-Fusion'ın global optimizasyon mekanizmasına dahil edilmesi, sistemin genel doğruluğunu ve tutarlılığını artırmayı hedefler. Yönelim ve irtifa (z) bileşenleri VIO tahminlerinden elde edilmeye devam edecek ve global optimizasyon sürecinde bu bilgiler de göz önünde bulundurulacaktır.

VIO, kamera ve IMU verilerini birleştirerek ego-hareketi tahmin eden güçlü bir tekniktir. VINS-Fusion gibi modern VIO sistemleri, durum tahmini için gelişmiş optimizasyon yöntemleri kullanır. Ancak, uzun süreli uçıuşlarda, özellikle çevresel özelliklerin tutarsız ölçümleri kaynaklı (rüzgar hızı kestirimi ve barometrik ölçüm kayıklıkları) veya GNSS sinyallerinin kesintili olduğu durumlarda, VIO tahminleri zamanla biriken sürüklenme hatalarına maruz kalır. AVL, mevcut bir kamera görüntüsünü önceden oluşturulmuş coğrafi referanslı bir haritayla eşleştirerek global konumu (örneğin, enlem ve boylam) belirleyebilir. Bu çalışma, AVL'den elde edilen yatay konum bilgilerini (enlem ve boylam) VINS-Fusion'a bir GNSS benzeri kaynak olarak entegre ederek bu sürüklenme sorununu hafifletmeyi amaçlamaktadır.

4.1.1 AVL Entegrasyonu

VINS-Fusion, sıkıca bağlı (tightly-coupled) bir yaklaşımla kamera ve IMU verilerini işleyen, optimizasyon tabanlı çoklu konumlandırma sensör desteğini içeren bir VIO sistemidir. Loop closure stratejisi olarak sürüklenmeyi düzeltmek

için daha önce ziyaret edilen yerleri algılar ve bir poz grafiği (pose graph) optimizasyonu gerçekleştirir (Bölüm 3.5.1). Buna ek olarak yüksek güvenilirlikli harici global konumlandırma verilerini (GNSS veya bu çalışmada AVL) bu global optimizasyon aşamasında entegre edebilir.

Önerilen sistemde, AVL algoritması tarafından üretilen enlem (Lat_{avl}) ve boylam (Lon_{avl}) bilgileri ile birlikte, k zamanında VIO tarafından tahmin edilen gövde çerçevesinin ENU koordinat sistemindeki z eksenine karşılık gelen irtifa bileşeni $(\mathbf{p}_{b_k}^{enu})z$ kullanılarak, $\mathbf{p}_{avl_gnss}^{enu}$ vektörü oluşturulur. Bu vektör, VINS-Fusion algoritmasına harici bir global konum ölçümü olarak entegre edilir. Yönelim bilgisi ile GNSS kaynaklı irtifa değeri ise, sistemin mevcut VIO tahminleri üzerinden elde edilmeye devam edilmektedir.

4.1.1.1 Görsel Konumlandırma Faktör Uyarlaması

VINS-Fusion' da bulunan mevcut 3D GNSS faktörü, AVL sonuçları (enlem ve boylam) konum bilgisini kullanacak şekilde alınmış, irtifa bilgisi VIO algoritmasında optimizasyonu etkilememesi için VIO algoritma çıktısından alınarak uyarlanmıştır.

AVL enlem ve boylam değerlerinin global optimizasyona GNSS kaynağı olarak beslenmesi için (Bölüm 3.5.1 de ifade edildiği üzere) $\mathbf{p}_{gnss}^{enu}$ ifadesi ve 3.40, 3.41, 3.42 denklemleri aşağıdaki gibi yeniden ifade edilebilir. VIO pozisyon bilgisinin ENU eksen takımı temsili (takeoff koordinatları ENU referans noktası kabul edilerek, $\mathbf{p}_{avl_gnss}^{enu}$) aşağıda yapılmıştır.

$$\mathbf{p}_{avl_gnss}^{enu} = \begin{bmatrix} (Lat_{avl})_x \\ (Lon_{avl})_y \\ \left(\mathbf{p}_{b_k}^{enu}\right)_z \end{bmatrix} \tag{4.1}$$

$$\hat{\mathbf{p}}_b^{enu} = \mathbf{R}_w^{enu}(\mathbf{p}_{b_k}^w + \mathbf{R}_{b_k}^w \mathbf{p}_b) + \mathbf{p}_w^{enu} \tag{4.2}$$

Elde edilen $\hat{\mathbf{p}}_b^{enu}$ vektörü, hava aracının body çerçevesine göre VIO pozisyon tahmininin ENU koordinat sistemindeki pozisyonunu ifade eder. GNSS faktörü ile ilişkili hata vektörü $\mathbf{r}_{avl_gnss,k}$ ise, tahmin edilen bu pozisyon ile AVL pozisyon ölçümü arasındaki farktan elde edilir ve aşağıdaki şekilde tanımlanır:

$$\mathbf{r}_{avl_gnss,k} = \hat{\mathbf{p}}_b^{enu} - \mathbf{p}_{avl_gnss}^{enu} \tag{4.3}$$

Benzer şekilde $\mathbf{r}_{avl_gnss,k}$ hata terimi, global optimizasyon problemine ilave edilerek güncellenmiş toplam maliyet fonksiyonu şu şekilde ifade edilir:

$$\min_{\mathcal{X},\mathbf{R}_w^{enu},\mathbf{p}_w^{enu}} \left(\sum_{\text{seq}} \|\mathbf{r}_{\text{seq}}\|^2 + \sum_{\text{loop}} \rho\left(\|\mathbf{r}_{\text{loop}}\|^2_{\Sigma_{loop}} \right) + \sum_{\text{avl_gnss}} \rho\left(\|\mathbf{r}_{\text{avl_gnss,k}}\|^2_{\Sigma_{avl_gnss}} \right) \right) \tag{4.4}$$

Global optimizasyon sonucunda, VIO yörüngesi AVL ölçümleriyle global olarak hizalanmış olurken, VIO'nun sağladığı yüksek frekanslı relatif hareket ve yönelim bilgisi tutarlılığı korunmuş olur.

4.1.1.2 AVL–VIO Entegrasyonunda Optimizasyon Parametreleri

AVL çıktılarının, Görsel-Ataletsel Odometri tabanlı VINS-Fusion algoritmasına global konumlandırma kaynağı olarak entegre edilmesi, görsel-ataletsel sistemlerde uzun vadede ortaya çıkan sürüklenme (drift) sorununu azaltmak açısından etkili bir yaklaşımdır. Bu entegrasyonun, sistemin yerel doğruluğu ve global tutarlılığı üzerindeki katkısının doğru şekilde değerlendirilebilmesi için bazı hiperparametrik faktörlerin dikkatle incelenmesi gerekmektedir.

AVL ölçümleri ile VIO tarafından tahmin edilen durum vektörleri arasındaki zaman senkronizasyonu, sistemin genel performansı açısından kritik öneme sahiptir. Zamanlama hataları veya gecikmeler, özellikle optimizasyon sürecinin doğruluğunu olumsuz etkileyebilir. Bu çalışmada, zaman farkı için kabul edilen maksimum tolerans değeri $500\,\text{ms}$ olarak belirlenmiştir.

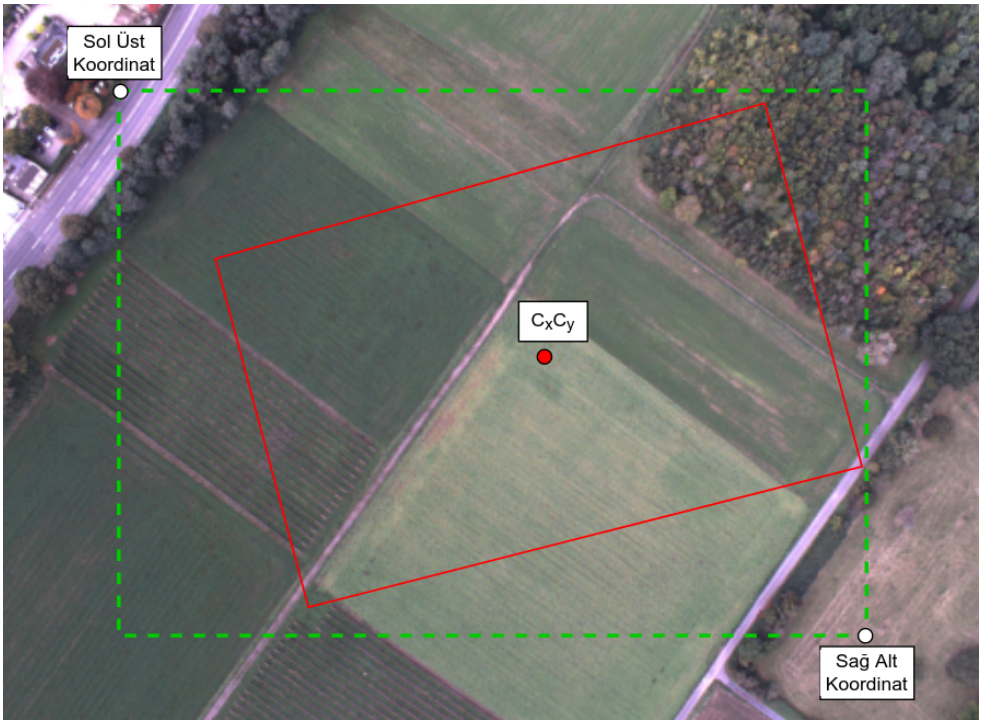
AVL yalnızca yatay konum (enlem ve boylam) bilgisi sağladığı için, sistemin yönelimi ve yükseklik (z ekseni) bileşeni VIO'nun IMU entegrasyonu ve görsel girdilerinden elde edilmeye devam etmektedir. Bu durum, optimizasyon sürecinin iki farklı kaynaktan gelen verileri tutarlı biçimde birleştirmesini zorunlu kılar. Özellikle barometrik veri kullanılmadığında, VIO'nun yükseklik tahminlerinde zamanla sürüklenme oluşabilir. AVL bu ekseni doğrudan düzeltmediği için, sistemin bu farkı entegrasyon aşamasında uyumlu şekilde yönetmesi önem taşır.

Sonuç olarak, AVL çıktılarının VINS-Fusion algoritmasına global konum girdisi olarak dahil edilmesi, VIO tabanlı sistemlerde uzun süreli sürüklenme sorununu azaltma potansiyeli taşıyan bir yöntemdir. AVL'den elde edilen enlem ve boylam verilerinin VINS-Fusion'ın global optimizasyon çerçevesine entegre edilmesi, sistemin hem yerel doğruluğunu hem de global tutarlılığını artırmaktadır. Bu

başarım, zaman senkronizasyonunun doğruluğu, koordinat dönüşümlerinin doğru yönetimi ve AVL ölçümlerinin güvenilirliğinin optimizasyona uygun biçimde yansıtılmasına bağlıdır.

4.1.2 AVL Konumunun Elde Edilmesi

Bu çalışmada, AVL yöntemleri kapsamında Gurgu ve arkadaşları [15] tarafından geliştirilen Wildnav modeli referans alınacaktır. Bu yöntem, GNSS sinyallerinin yetersiz olduğu durumlarda İHA'ların konumlarını belirlemek amacıyla, kamera görüntülerinin coğrafi olarak referanslanmış uydu görüntüleriyle eşleştirilmesine dayanmaktadır. Çalışmanın detayları bu bölümde ayrıntılı olarak ele alınacaktır.



Şekil 4.1 Referans hava aracı görseli ile temsili eşleme alanı [49]

Wildnav çalışması referans alındığında, uçuş bölgesine ait yüksek çözünürlüklü uydu görüntüleri açık kaynaklardan temin edilmiş ve her bir görüntü kesiti (tile), köşe noktalarına karşılık gelen enlem ve boylam bilgileriyle coğrafi olarak referanslandırılmıştır. Bu referanslandırma sayesinde, görüntüdeki herhangi bir pikselin jeodezik koordinatlarına dönüştürülmesi mümkün hâle gelmektedir.

Şekil 4.1'de, yeşil çizgiler ilgili harita parçasının sınırlarını, kırmızı çizgiler ise eşleme işleminden sonra elde edilen perspektif dönüşüm sınırlarını temsil etmektedir. Görüntü kesiti içerisindeki piksel koordinatları (C_x, C_y) ile coğrafi koordinatlar arasındaki doğrusal dönüşüm ilişkisi ise aşağıda verilmiştir:

$$\text{Lat} = \text{Lat}_t + \frac{C_y}{H} \times (\text{Lat}_b - \text{Lat}_t) \tag{4.5}$$

$$\text{Lon} = \text{Lon}_t + \frac{C_x}{W} \times (\text{Lon}_b - \text{Lon}_t) \tag{4.6}$$

Burada $(\text{Lat}_t, \text{Lon}_t)$ sol üst köşe, $(\text{Lat}_b, \text{Lon}_b)$ sağ alt köşe koordinatlarını; H ve W ise kesitin piksel boyutlarını temsil eder.

İHA tarafından yakalanan görüntüler ile veri tabanındaki uydu görüntü kesitleri karşılaştırıldığında, her iki görüntü türünden de belirgin görsel özellikler çıkarılmaktadır. Bu eşleştirme işlemi için, farklı perspektif ve aydınlatma koşullarında dahi yüksek eşleştirme doğruluğu sunabilen, derin öğrenme tabanlı SuperGlue modeli [50] kullanılmaktadır. SuperGlue, gerçek dünya görüntüleri ile harita temelli görüntüler arasında anahtar noktaların ve görsel tanımlayıcıların güvenilir bir şekilde eşleştirilmesini sağlayarak, mevsimsel değişiklikler gibi zorlu çevresel koşullarda bile başarılı sonuçlar elde edilmesine olanak tanımaktadır.

En iyi eşleşen uydu görüntüsü kesiti belirlendikten sonra, bu eşleşmeye ait ortak noktalardan yararlanılarak iki görüntü arasındaki izdüşümsel dönüşüm, yani homografi matrisi hesaplanır. Bu aşamada, hatalı eşleşmelerin elenmesi (outlier elimination) amacıyla RANSAC (Random Sample Consensus) algoritması [51] kullanılmaktadır.

Hesaplanan homografi matrisi, İHA görüntüsünde seçilen bir bölgenin uydu görüntüsü kesiti üzerindeki karşılığının belirlenmesini sağlar. Uydu görüntüsü üzerinde elde edilen bu bölgeye ait piksel koordinatları, Eşitlik 4.5 ve 4.6 aracılığıyla coğrafi koordinatlara dönüştürülür. Böylece, İHA'nın bulunduğu konum, uydu görüntü veritabanı üzerinden yapılan en güvenilir eşleşmeye dayanarak kestirilmiş olur.

4.2 Tezde Ele Alınan AVL Geliştirmeleri

Bu yüksek lisans tezinde, Wildnav çalışmasının ortaya koyduğu temeller üzerine inşa edilerek, özellikle aşağıdaki hususların daha detaylı irdelenmesi ve iyileştirmeler sunulması hedeflenmektedir:

SuperGlue modelinin eğitim verisinin sınırlı olması nedeniyle sadece belli rotasyon açılarına kadar başarılı eşleştirme yapabildiği gözlemlenmiştir. Bu yüzden, İHA görüntülerinin kuzeye yönlendirilmiş uydu haritalarıyla doğru şekilde hizalanabilmesi için İHA'nın baş açısı (yaw) bilgisinin kullanılması gerekmektedir.

Ayrıca, Wildnav çalışmasında İHA'nın yaklaşık 120 metre yükseklikte uçtuğu ve uydu görüntülerinin de benzer bir görüş alanı sunacak şekilde seçildiği belirtilmiştir. Ancak, İHA'nın uçuş irtifası ile referans uydu görüntülerinin ölçeği arasındaki farklılıkların eşleme performansı üzerindeki etkisi sistematik olarak incelenmemiştir. Bu tezde, farklı İHA uçuş irtifalarından elde edilen görüntülerin, sabit ölçekli bir uydu haritasıyla nasıl etkin bir şekilde eşleştirilebileceği ve bu ölçek farklılıklarının yerelleştirme doğruluğuna etkileri detaylı bir şekilde analiz edilecektir. Bu kapsamda, ölçek değişimlerine karşı daha dayanıklı özellik tanımlayıcıları ve eşleme yöntemleri veya dinamik olarak farklı ölçeklerdeki harita kesitlerini kullanabilen bir yaklaşım geliştirilmesi hedeflenmektedir.

4.2.1 Manyetometre Destekli Görsel Dönmelerin Telafisi

İHA' nın uçuş esnasındaki yaw açısındaki değişimleri, kuzey yönüne referanslanmış uydu görselleri arasında önemli yönelim farklılıkları oluşturarak SuperGlue gibi gelişmiş eşleştiricilerin dahi performansını olumsuz etkilemektedir. Bu noktada, manyetometre verisinden elde edilen yaw açısı ile görüntüyü referans oryantasyona döndürerek SuperGlue eşleştiricisinin yönelim değişimlerine karşı hassasiyetini azaltmayı ve bu sayede eşleme performansını artırmak amaçlanmıştır.

Ardışık iki İHA görüntüsü arasındaki göreceli yaw farkını telafi etmek amacıyla, mevcut İHA görüntüsü, manyetometreden elde edilen yaw açısı (θ_{yaw}) kadar, görüntü merkezi etrafında döndürülür. Bir (x, y) noktasının, orijin etrafında θ açısı kadar saat yönünün tersine döndürülerek yeni (x', y') koordinatlarına dönüşümü şu şekilde ifade edilir:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{4.7}$$

Görüntü döndürüldüğünde, orijinal görüntü sınırlarının dışına taşan piksellerin kaybolmaması için döndürülmüş görüntüyü tamamen içerecek yeni sınırlayıcı kutu boyutları (w_{new}, h_{new}) hesaplanır:

$$\begin{aligned} \cos_{\text{abs}} &= |\mathbf{M}[0,0]| = |\cos(\theta_{\text{yaw}})| \\ \sin_{\text{abs}} &= |\mathbf{M}[0,1]| = |\sin(\theta_{\text{yaw}})| \\ w_{\text{new}} &= \text{int}((h \cdot \sin_{\text{abs}}) + (w \cdot \cos_{\text{abs}})) \\ h_{\text{new}} &= \text{int}((h \cdot \cos_{\text{abs}}) + (w \cdot \sin_{\text{abs}})) \end{aligned}$$

Döndürme matrisi M, bu yeni boyutlara göre görüntüyü ortalamak amacıyla güncellenir. Bu, dönme merkezinin yeni görüntü alanının merkezine kaydırılması anlamına gelir:

$$\begin{aligned} \text{shift}_x &= (w_{\text{new}}/2) - c_x \\ \text{shift}_y &= (h_{\text{new}}/2) - c_y \\ M[0,2] &\leftarrow M[0,2] + \text{shift}_x \\ M[1,2] &\leftarrow M[1,2] + \text{shift}_y \end{aligned}$$

Bu güncellenmiş M matrisi ile görüntü döndürülür ve orijinal görüntü köşe koordinatları yeniden hizalandırılır.

İHA'nın , SuperGlue'nun doğrudan başa çıkmakta zorlanacağı yönelim farklılıkları yaratır. Önerilen yöntem, dinamik manevraları sonucu oluşabilecek büyük yaw değişimlerini (örneğin, 90° veya üzeri), manyetometreden elde edilen $\theta_{\text{yaw_ölçülen}}$ ile görüntüyü önceden döndürerek, iki görüntü arasındaki göreceli açıyı SuperGlue'nun daha etkin çalışabileceği bir aralığa (örneğin, $\pm 20 - 30^\circ$) indirmeyi hedefler. Bu, bir "kaba hizalama" işlevi görür. Bu kaba hizalama ile görüntüler birbirine daha iyi hizalandığında, SuperGlue daha fazla sayıda ve daha güvenilir geometrik tutarlılığa sahip eşleşme noktası bulabilmektedir.

Bu yaklaşım, SuperGlue'nun doğasında var olan yönelim limitlerine karşı pratik bir çözüm sunarak, zorlu koşullarda görüntü eşleme doğruluğunu artırmaya yönelik önemli bir katkıdır.

4.2.2 Görüş Alanına Bağlı Çözünürlük Dengelemesi

İHA ile elde edilen hava fotoğraflarının mevcut harita verileriyle eşlenmesi ve bu süreçte ortaya çıkan çözünürlük farklılıklarının yönetilmesi, eşleme performansı açısından kritik öneme sahiptir. Bu noktada, İHA'nın uçuş irtifası ve kamera özelliklerinden yararlanarak, yeryüzünde taradığı alanın geometrik olarak hesaplanması ve bu alanın referans harita üzerindeki piksel bazında karşılığının belirlenerek çözünürlük dengelemesi yapılmıştır.

4.2.2.1 Yeryüzünde Taranan Alanın Hesaplanması

İHA'nın metre bazında H uçuş irtifasında (Above Ground Level, AGL) seyrederken, üzerinde bulunan kamera ile yeryüzünde taradığı alanın boyutları,

kameranın sahip olduğu yatay (FOV_h) ve dikey (FOV_v) görüş açılarına bağlı olarak değişmektedir. Taranan alanın yeryüzündeki yatay (D_h) ve dikey (D_v) uzunlukları, metre cinsinden, temel trigonometrik prensipler kullanılarak aşağıdaki gibi ifade edilebilir:

$$D_h = 2 \cdot H \cdot \tan\left(\frac{FOV_h}{2}\right) \tag{4.8}$$

$$D_v = 2 \cdot H \cdot \tan\left(\frac{FOV_v}{2}\right) \tag{4.9}$$

Bu denklemlerde H, İHA'nın yerden yüksekliğini (metre); FOV_h ve FOV_v ise kameranın sırasıyla yatay ve dikey görüş açılarını temsil etmektedir. Bu hesaplamalar neticesinde elde edilen D_h ve D_v değerleri, İHA'nın tek bir görüntü karesi ile yeryüzünde kapladığı dikdörtgenin metre ölçeğinde yatay ve dikey mesafelerini verir.

4.2.2.2 Taranan Alanın Harita Üzerindeki Piksel Karşılığının Belirlenmesi

Yeryüzünde hesaplanan bu alanın, referans olarak kullanılan harita üzerinde hangi piksel bölgesine denk geldiğinin tespiti, haritanın mekansal çözünürlüğü ile doğrudan ilişkilidir. Haritanın Yer Örnekleme Mesafesi (Ground Sample Distance, GSD), GSD_{map}, harita üzerindeki tek bir pikselin yeryüzünde kaç metrelik bir uzunluğa karşılık geldiğini ifade eder (örneğin, 0.5 m/piksel). İHA tarafından taranan D_h ve D_v boyutlarındaki yeryüzü alanının harita üzerindeki piksel cinsinden eni (P_h) ve boyu (P_v) aşağıdaki formüllerle hesaplanır:

$$P_h = \frac{D_h}{GSD_{map}} \tag{4.10}$$

$$P_v = \frac{D_v}{GSD_{map}} \tag{4.11}$$

Bu hesaplamalar sonucunda bulunan P_h ve P_v değerleri, harita üzerinden kesilerek (crop edilerek) alınacak olan dikdörtgen bölgenin piksel cinsinden boyutlarını tanımlar. Görüntü eşleme aşamasında, İHA görüntüsünün merkezi ile bu hesaplanan harita alanının merkezinin çakıştırılması esastır.

4.2.3 Yönelim Tabanlı Konum İyileştirmesi

AVL algoritması ile elde edilen hedef noktasının coğrafi koordinatlarından, hava aracının kendi konumunu geri hesaplamak için bir pozisyon düzeltme yöntemi geliştirilmiştir. Bu süreç, hava aracının bilinen bir hedef noktasına bakarken sahip olduğu anlık duruş bilgilerini ve yerden yüksekliğini (AGL) kullanarak, hedefe göre olan geometrik konumunu çözümler. Bu sayede, gözlemlenen hedef koordinatları, hava aracının küresel konumunu belirlemek için bir referans noktasına dönüştürülür.

Literatürde bu yöntem poz kestirimi (pose estimation) olarak tanımlanmakta olup, yönelime bağlı olarak hava aracının uzaydaki konumunu doğru biçimde belirlemeyi amaçlamaktadır. Bu kapsamda, çalışmada kullanılan dönüşüm işlemleri için belirtilen referans sistem çerçeveleri aşağıda tanımlanmıştır:

- **Jeodezik Koordinat Sistemi**: Konumların enlem (ϕ), boylam (λ) ve WGS84 elipsoidine göre yükseklik (h) ile ifade edildiği küresel sistemdir.
- **Gövde Koordinat Sistemi ($\mathcal{F}_b$)**: Başlangıç noktası hava aracının ağırlık merkezi olan, araca sabitlenmiş bir kartezyen sistemdir. Eksenleri X ekseni ileri, Y ekseni sola ve Z ekseni yukarı yönü gösterecek şekilde tanımlanmıştır. Kamera, bu sistemin $-z$ ekseni boyunca aşağıya bakacak şekilde monte edilmiştir.
- **Yerel Seyrüsefer Koordinat Sistemi ($\mathcal{F}_n$)**: Hesaplamaların yapıldığı, başlangıç noktası yeryüzündeki hedef noktası olan yerel bir kartezyen sistemdir. ENU eksen takımını referans alır.

Hava aracının konumunun düzeltilmesi süreci, gözlem vektörünün tanımlanması, duruş açılarından elde edilen dönüşüm matrislerinin oluşturulması ve bu dönüşüm sonucunda elde edilen yer değiştirme vektörünün coğrafi koordinat sistemine aktarılması adımlarını kapsamaktadır.

İlk aşamada, hava aracından hedefe yönelen vektör, gövde koordinat sisteminde ($\mathcal{F}_b$) ifade edilir. Hava aracının hedefin tam üzerinde bulunduğu ve yer seviyesinden h_{AGL} metre yüksekte olduğu varsayıldığında, kamera doğrudan aşağıya bakacak şekilde tanımlandığında bu vektör şu şekilde yazılır:

$$\mathbf{v}_b = \begin{bmatrix} 0 \\ 0 \\ -h_{AGL} \end{bmatrix} \tag{4.12}$$

Hava aracının duruşu Euler açıları (yaw ψ, pitch θ ve roll ϕ) ile temsil edilmektedir. Bu açılar, gövde koordinat sisteminden ($\mathcal{F}_b$) yerel seyrüsefer sistemine ($\mathcal{F}_n$) geçişi sağlayan dönüşüm matrisinin oluşturulmasında kullanılmaktadır. Dönüşüm, Z–Y–X (yaw–pitch–roll) sıralamasıyla gerçekleştirilmektedir:

$$\mathbf{C}_n^b = \mathbf{R}_z(\psi)\,\mathbf{R}_y(\theta)\,\mathbf{R}_x(\phi) \tag{4.13}$$

Burada $\mathbf{R}_z(\psi)$, $\mathbf{R}_y(\theta)$ ve $\mathbf{R}_x(\phi)$ sırasıyla Z, Y ve X eksenleri etrafındaki temel dönüşüm matrislerini ifade etmektedir.

Tanımlanan bu dönüşüm matrisi yardımıyla, gövde koordinat sisteminde bulunan gözlem vektörü $\mathbf{v}_b$, yerel seyrüsefer sistemine dönüştürülmekte ve hava aracından hedefe olan yer değiştirme $\mathbf{v}_n$ elde edilmektedir:

$$\mathbf{v}_n = \mathbf{C}_n^b\,\mathbf{v}_b = \begin{bmatrix} d_E \\ d_N \\ d_U \end{bmatrix} \tag{4.14}$$

Bu vektörün yatay bileşenleri (d_E, d_N), metre cinsinden doğu ve kuzey yönlerindeki yer değiştirmeyi; d_U bileşeni ise düşey yer değiştirmeyi ifade etmektedir. Hava aracının coğrafi konumu, bilinen hedef konumuna bu yer değiştirme vektörünün negatifinin eklenmesiyle hesaplanmaktadır.

Yatay konum (enlem ve boylam) için dönüşüm, Dünya'nın yarıçapı R_e dikkate alınarak aşağıdaki şekilde yapılmaktadır:

$$\phi_a = \phi_t + \left(\frac{-d_N}{R_e}\right)\left(\frac{180}{\pi}\right) \tag{4.15}$$

$$\lambda_a = \lambda_t + \left(\frac{-d_E}{R_e\cos(\phi_t \cdot \pi/180)}\right)\left(\frac{180}{\pi}\right) \tag{4.16}$$

Dikey konum ise, hedefin bilinen rakımına $-d_U$ değerinin eklenmesiyle hesaplanmaktadır:

$$h_a = h_t - d_U \tag{4.17}$$

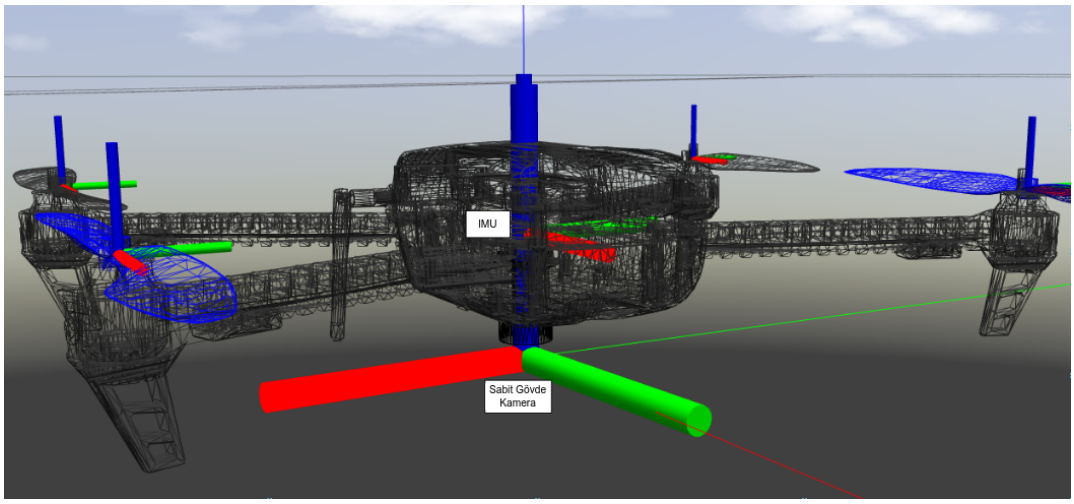
Sonuç olarak, hedef noktasının bilinen konumu ve hava aracının anlık duruş bilgileri

birlikte kullanılarak, hava aracının üç boyutlu coğrafi konumu hassas biçimde kestirilebilmektedir.

5
DENEYSEL ÇALIŞMALAR VE SONUÇLAR

5.1 Simülasyon Ortamı

5.1.1 Iris Hava Aracı Model Analizi



Şekil 5.1 Simülasyon ortamında tanımlanan Iris çok rotorlu hava aracı

Bu bölümde, tez kapsamında kullanılan Gazebo simülasyon ortamı için geliştirilmiş olan Iris hava aracı modelinin analizi gerçekleştirilmektedir. İnceleme; hava aracının fiziksel, kinematik, sensör ve dinamik özelliklerini kapsamakta olup, otonom uçuş algoritmalarının tasarımı için gerekli olan temel parametreleri ortaya koymaktadır.

5.1.1.1 Sensör Donanımı ve Özellikleri

Hava aracı, otonom seyrüsefer ve çevre algılama görevleri için kritik olan monoküler kamera ve IMU sensörleri donatılmıştır.

Gövde altından bakacak şekilde konumlandırılan monoküler kameranın temel özellikleri şunlardır:

- **Görüş Açısı (Field of View, FoV):** Gövde kameranın yatay görüş açısı

$2.0944\,\mathrm{rad}$ olarak belirtilmiştir. Bu değer, derece cinsinden yaklaşık olarak $120°$'ye tekabül eder ve algoritmaların geniş bir alanı taramasına olanak tanır. Kamera, 752x480 piksel çözünürlükte ve L8 (8-bit grayscale) formatında görüntü sağlamaktadır.

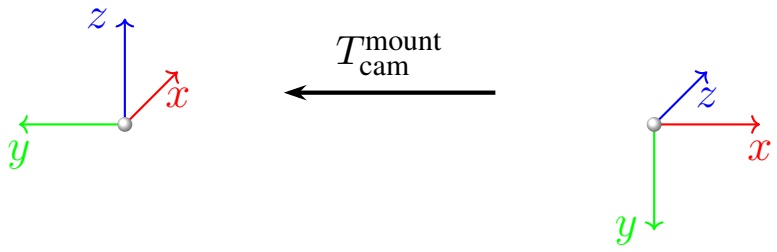
- **Çalışma Frekansı:** Sensörün frekansı $(20\,\mathrm{Hz})$ olarak ayarlanmıştır. Bu frekans, yüksek hızda hareket etmeyen hava araçları için görsel odometri ve SLAM uygulamalarında yeterli bir hızdır.

IMU, hava aracının açısal hızlarını ve doğrusal ivmelerini ölçerek durum kestirimi algoritmaları için temel verileri sağlar. ArduPilot uçuş kontrol yazılımıyla entegrasyon için özel olarak yapılandırılmıştır ve 1000.0 Hz frekansında çalışır. Bu yüksek frekans, özellikle agresif manevralar sırasında ve yüksek hızlı kontrol döngülerinde aracın durumunun hassas bir şekilde takip edilebilmesini sağlar. Bu IMU'nun `base_link` koordinat çerçevesi ile aynı pozisyonda olup, ArduPilot'un kullandığı NED (North-East-Down) koordinat sistemiyle uyumluluğu sağlamak amacıyla X ekseni etrafında 3.141593 radyan dönüklük uygulanarak otopilota aktarılmaktadır.

5.1.2 Gövde ve Kamera Arasındaki Kinematik Dönüşüm

Otonom uçuşta, sensör verilerinin aracın gövde (body) koordinat sistemine dönüştürülmesi esastır. Bu dönüşüm, aracın ana referans çerçevesi olan `base_link` ile kamera sensörünün referans çerçevesi arasındaki ilişkiyi tanımlayan homojen bir transformasyon matrisi ile ifade edilir. Kameranın `base_link`'e göre konumu ve yönelimi, `<0, 0, -0.05, 0, 1.5708, 0>` olarak tanımlanmıştır. Bu tanım, sırasıyla (x, y, z) ötelemesini ve (roll, pitch, yaw) Euler açılarını belirtir.

- **Gövde Koordinat Sistemi (F_{body}):** X ekseni ileri, Y ekseni sol, Z ekseni yukarı yönelimli sağ-elli bir sistemdir. Gövde pozu, dünya (world) çerçevesi eksen takımları aynı kabul edilmiştir ($T_{\text{body}}^{\text{world}} = I$).
- **Kamera Koordinat Sistemi (F_{cam}):** Z ekseni ileri, X ekseni sağ, Y ekseni aşağı yönelimli sol-elli bir optik sistemdir.
- **Kamera Pozu:** Gövdeye göre kamera pozisyonu $(x, y, z) = (0, 0, -0.05)$ ve oryantasyonu (Roll, Pitch, Yaw) = $(0, 1.5708, 0)$ radyan olarak verilmiştir. Bu, Y ekseni etrafında $\pi/2$ radyanlık bir Pitch dönüşüne karşılık gelir.



Şekil 5.2 Kamera ve Body Mount eksen takımları arasındaki dönüşüm

5.1.3 Dönüşüm Metodolojisi

Nihai dönüşüm matrisi ($T_{\text{cam}}^{\text{body}}$), kameranın gövdeye göre dışsal (extrinsic) pozunun ($T_{\text{mount}}^{\text{body}}$) ve kamera eksenlerinin standart eksenlere göre içsel (intrinsic) dönüşümünün ($T_{\text{cam}}^{\text{mount}}$) çarpımı ile elde edilir.

$$T_{\text{cam}}^{\text{body}} = T_{\text{mount}}^{\text{body}} \cdot T_{\text{cam}}^{\text{mount}} \tag{5.1}$$

5.1.3.1 Eksen Dönüşüm Matrisi ($T_{\text{cam}}^{\text{mount}}$)

$T_{\text{cam}}^{\text{mount}}$ matrisi, kameranın optik eksen takımını (F_{cam}), mount eksen takımına (Gazebo World frame eksen takımı, F_{mount}) olan dönüşümü temsil etmektedir.

$$T_{\text{cam}}^{\text{mount}} = \begin{pmatrix} & R_{\text{pose}} & & t_{\text{pose}} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.2}$$

5.1.3.2 Poz Dönüşüm Matrisi ($T_{\text{mount}}^{\text{body}}$)

Verilen öteleme ve dönme bilgisinden elde edilen dışsal poz matrisidir. Y ekseni etrafında $\theta = \pi/2$ radyanlık bir dönüş (R_{pose}) ve $(0, 0, -0.05)$ metrelik bir öteleme (t_{pose}) içerir. Dönüş matrisi, genel $R_y(\theta)$ formülü kullanılarak hesaplanır:

$$R_{\text{pose}} = R_y(\pi/2) = \begin{pmatrix} \cos(\pi/2) & 0 & \sin(\pi/2) \\ 0 & 1 & 0 \\ -\sin(\pi/2) & 0 & \cos(\pi/2) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix} \tag{5.3}$$

Bu rotasyon ve öteleme, aşağıdaki homojen dönüşüm matrisini oluşturur:

$$T_{\text{mount}}^{\text{body}} = \begin{pmatrix} & R_{\text{pose}} & & t_{\text{pose}} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -0.05 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.4}$$

5.1.3.3 Kameradan Gövdeye Dönüşüm ($T_{\text{cam}}^{\text{body}}$)

Yukarıdaki matrislerin çarpımı ile nihai dönüşüm matrisi elde edilir:

$$T_{\text{cam}}^{\text{body}} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -0.05 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.5}$$

$$\mathbf{T}_{\text{cam}}^{\text{body}} = \begin{pmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -0.05 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.6}$$

Denklem 5.6, kameradan gövdeye dönüşüm matrisini göstermektedir.

5.1.4 Rotor Bilgileri ve Uçuş Performansı

Hava aracının kalkış ve hareket kabiliyeti, motorların gücü, pervane aerodinamiği ve toplam kütle gibi faktörlere bağlıdır. Modelin toplam kütlesi, ana gövde ($1.500\,01\,\text{kg}$), dört rotor (toplam $0.02\,\text{kg}$) ve IMU sensörleri (toplam $0.165\,\text{kg}$) gibi ana bileşenlerin kütlelerinin toplamından oluşur ve yaklaşık olarak $m_{\text{total}} \approx 1.685\,\text{kg}$'dır.

Hava aracının havada asılı kalabilmesi (hover) için dört rotorun ürettiği toplam itki kuvvetinin (F_{thrust}), yerçekimi kuvvetini (F_{gravity}) dengelemesi gerekir.

$$F_{\text{thrust}} = F_{\text{gravity}} = m_{\text{total}} \times g \approx 1.685\,\text{kg} \times 9.81\,\text{m}\,\text{s}^{-2} \approx 16.53\,\text{N} \tag{5.7}$$

Her bir motor için itki kuvveti, $F_{\text{motor}} = k_m \cdot \omega^2$ formülü ile modellenir. Burada k_m motor sabitini ($8.548\,58 \times 10^{-6}$), ω ise rotorun açısal hızını ($\text{rad}\,\text{s}^{-1}$) temsil eder. Havada asılı kalma durumu için her bir rotorun sahip olması gereken ortalama açısal hız (ω_{hover}) şu şekilde hesaplanabilir:

$$4 \times (8.548\,58 \times 10^{-6}) \times \omega_{\text{hover}}^2 = 16.53\,\text{N} \tag{5.8}$$

$$\omega_{\text{hover}} = \sqrt{\frac{16.53}{4 \times 8.548\,58 \times 10^{-6}}} \approx 695\,\text{rad}\,\text{s}^{-1} \tag{5.9}$$

Motorların maksimum açısal hızı $1100\,\text{rad}\,\text{s}^{-1}$ olarak tanımlanmıştır. Hesaplanan ω_{hover} değerinin, bu maksimum hızın altında olması, hava aracının tırmanma ve yanal manevralar için yeterli güç rezervine sahip olduğunu göstermektedir.

5.2 Uçuş Görev Planları

Simülasyon ortamında gerçekleştirilen çalışmalarda, çembersel ve üçgensel biçimlerde tasarlanmış farklı uçuş planları uygulanmıştır. Belirlenen uçuş planları, geliştirilen yöntemin yörünge takibi performansını ve zamanla oluşan shifting durumlarına karşı dayanıklılığını değerlendirmek üzere yürütülmüştür.

5.2.1 Üçgensel Form Uçuşu

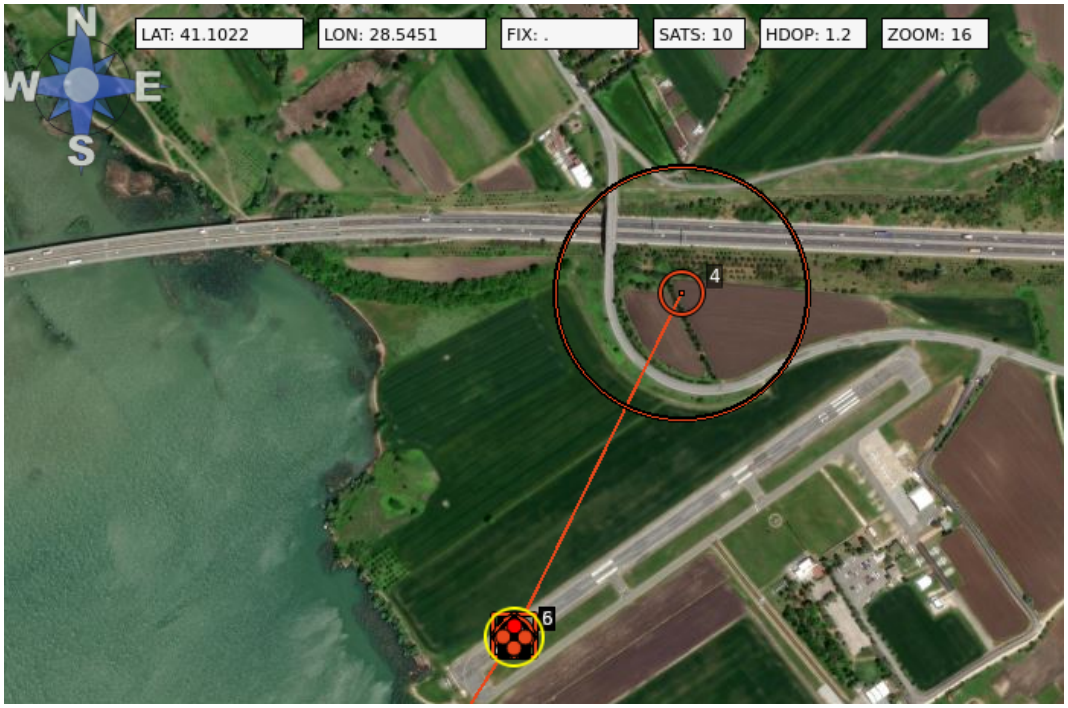


Şekil 5.3 Üçgensel form uçuş planı

Üçgensel form uçuş planı, İHA'nın kalkış noktasından hareket ederek yaklaşık 2.178 m uzunluğundaki yatay rotayı farklı irtifalarda tamamlamasını kapsamaktadır. Bu rota, 100 m ve 300 m relatif irtifalarda gerçekleştirilmiş

olup, uçuşun sonunda İHA başlangıç noktasına geri dönerek iniş yapmaktadır. Bu senaryo, VIO algoritmasının düz uçuş esnasındaki performansını ve İHA'nın kendi ekseni etrafında dönme (yaw) hareketi sırasında yörünge takibi doğruluğunu değerlendirmek amacıyla tasarlanmıştır.

5.2.2 Çembersel Form Uçuşu



Şekil 5.4 Çembersel form uçuş planı

Çembersel uçuş planı kapsamında, ilk senaryoda 150 m yarıçaplı bir çember etrafında sırasıyla 100 m, 150 m ve 200 m irtifalarda; ikinci senaryoda ise 200 m yarıçaplı bir çember etrafında 300 m, 400 m ve 350 m irtifalarda uçuşlar gerçekleştirilmiştir. Her bir uçuş senaryosu, İHA'nın belirlenen çember hattı üzerinde üç tam tur atacak şekilde planlanmıştır. Bu uçuşlar, VIO algoritmasının farklı irtifa ve yörünge koşullarındaki loop-closure kararlılığını değerlendirmek amacıyla tasarlanmıştır.

Tablo 5.1 Uçuş planları özet tablosu

Sekans	İrtifa (m, Rel.Alt)	Uçuş Mesafesi (m)	Tur Sayısı
100m_iris_triangle	100	2378	1
300m_iris_triangle	300	2757	1
100m_iris_circle	100 - 150 - 200	3703	3
300m_iris_circle	300 - 400 - 350	4809	3

Sonuç olarak, bu uçuş planları; GNSS'siz seyrüsefer algoritmalarının IMU

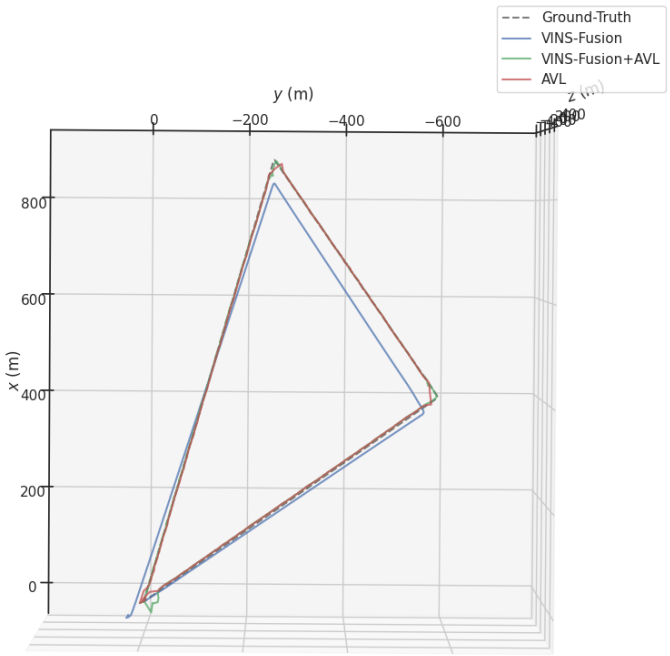
kaymalarını telafi etme, farklı irtifa ve çevresel koşullarda görsel özellikleri güvenilir biçimde eşleme ve uzun vadeli konum tahminini kararlı şekilde sürdürebilme kabiliyetlerini bütüncül olarak değerlendirmek amacıyla uygun ve zorlu bir test senaryosu sunmaktadır.

5.3 Simülasyon Ortam Sonuçları

Bu bölümde, önerilen **VINS-Fusion + AVL** yaklaşımının, temel **VINS-Fusion** algoritmasına kıyasla sağladığı performans iyileştirmeleri, simülasyon ortamında elde edilen uçuş verileri üzerinden analiz edilmiştir.

5.3.1 Üçgensel Form Uçuş Analizi

Tablo 5.1 de belirtilen irtifa ve uçuş planları referans alınarak *100m_iris_triangle* uçuşunun test sonucu, Şekil 5.5 üzerinden gözlemlenebilir.



Şekil 5.5 100m üçgensel form uçuşu

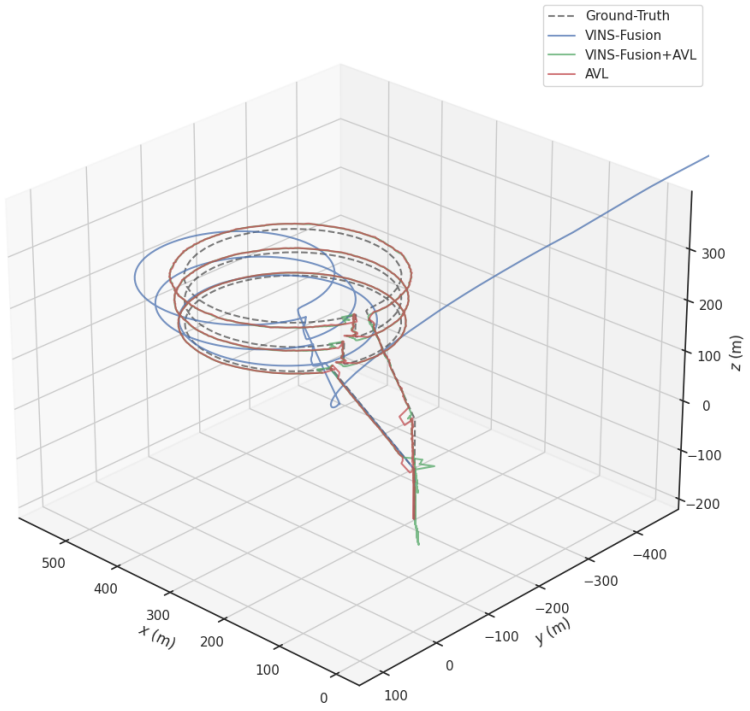
Şekil 5.5'te sunulan 100m' lik üçgensel form uçuşunun yörünge sonuçları, önerilen AVL modülünün VINS-Fusion performansı üzerindeki etkisini nitel olarak göstermektedir.

Standart VINS-Fusion algoritmasının ürettiği yörünge, genel üçgensel yapıyı takip edebilmekle birlikte, düz yörünge hatları boyunca belirgin bir birikimli ölçek ve pozisyon sapması sergilemektedir. Bu sapma, kestirilen yörüngenin

referans yörüngeye kıyasla hem daha küçük kalması hem de yörüngenin sonlanma noktasının başlangıç noktasından belirgin şekilde ayrılmasıyla kendini göstermektedir. Bu durum, yalnızca görsel ve ataletsel verilere dayalı sistemlerin, özellikle uzun mesafeli görevlerde birikimli hatalara ve ölçek belirsizliğine ne kadar açık olduğunun tipik bir örneğidir.

Buna karşın, önerilen VINS-Fusion + AVL yaklaşımının yörüngesi, referans yörüngeye AVL Faktör etkisiyle çok daha yakın kalmaktadır. Grafikte de görüldüğü gibi, önerilen metodolojinin ürettiği yörünge, referans verisiyle neredeyse tam bir örtüşme sergilemektedir. AVL modülünden gelen mutlak konum verilerinin, VINS-Fusion'ın birikimli sapmasını sürekli olarak düzeltmek suretiyle birincil stabilizasyon mekanizması olarak çalıştığı anlaşılmaktadır. Bu sayede, temel VINS-Fusion'da gözlemlenen ölçek hatası ve pozisyonel sürüklenmesi etkin bir şekilde bastırılmıştır.

5.3.2 Çembersel Form Uçuş Analizi



Şekil 5.6 100m çembersel form uçuşu

Şekil 5.6, AVL modülünün VIO sistemi üzerindeki stabilize edici etkisinin en belirgin olduğu 100m' lik çembersel form uçuşu senaryosunu görselleştirmektedir. Bu zorlu senaryo, tekrarlı ve dinamik hareketlerin VIO sistemleri üzerindeki birikimli sapma etkisini test etmek için ideal bir ortam sunmaktadır.

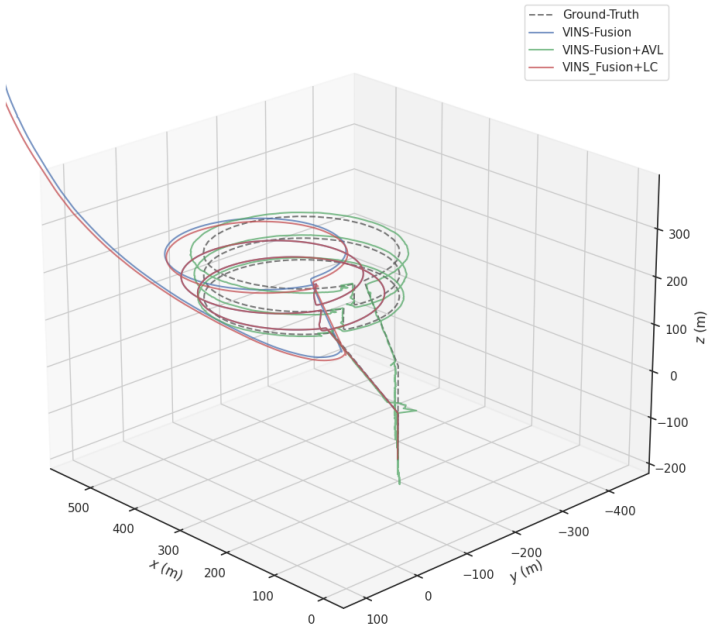
Temel VINS-Fusion algoritmasının ürettiği yörünge görevin başlangıcında referans verisini bir süre takip etse de, sistemin tekrarlı ve karmaşık bir manevra olan dairesel yörüngeye girmesiyle birlikte catastrophic drift yaşamaktadır. Algoritma, birikimli hatalar nedeniyle dairesel yörüngenin yapısını koruyamamakta, çemberlerin yarıçapı ve konumu giderek bozulmakta ve manevra sonrasında yörüngeden tamamen koparak sonsuza doğru uzaklaşmaktadır. Bu durum, standart bir VIO sisteminin uzun süreli ve görsel olarak tekrarlı görevlerdeki doğal zayıflığını ve başarısızlığını açıkça göstermektedir.

Buna karşın, önerilen VINS-Fusion + AVL yaklaşımı, tüm görev boyunca referans yörüngeye bağlı kalmaktadır. İki yaklaşım arasındaki en kritik fark, dairesel uçuştan sonraki doğrusal kısımda ortaya çıkmaktadır: Temel VINS-Fusion'da gözlemlenen ciddi sürüklenme hatası, önerilen sistemde etkin bir şekilde tamamen bastırılmış ve sistem, görev sonuna kadar tutarlılığını ve yörünge takibini başarıyla sürdürmüştür.

Tablo 5.2 VINS-Fusion + AVL çalışma başarımının, referans VINS-Fusion algoritması ile simülasyon uçuş sekansları üzerindeki karşılaştırmalı sonuçları

	VINS-Fusion		VINS-Fusion + AVL	
Sekans	ATE (m)	Drift [%]	ATE (m)	Drift [%]
100m_iris_triangle	37.91	1.594	**3.78**	**0.159**
100m_iris_circle	280.12	7.563	**11.78**	**0.318**
300m_iris_triangle	573.60	20.801	**17.57**	**0.637**
300m_iris_circle	1119.43	23.276	**73.97**	**1.538**

Tablo 5.2'de sunulan sonuçlarda, uçuş yörüngesinin ölçeği ve karmaşıklığı arttıkça, önerilen VINS-Fusion + AVL metodolojisinin sağladığı performans kazanımının katlanarak arttığı net bir şekilde gözlemlenmektedir. Temel VINS-Fusion algoritmasının Mutlak Yörünge Hatası (Absolute Trajectory Error, ATE), en basit senaryo olan *100m_iris_triangle*'da 37.91 metreden başlayarak en zorlu senaryo olan *300m_iris_circle*'da 1119.43 metrelik başarısızlık durumuna ulaşırken, AVL ile güçlendirilmiş sistem hatayı tutarlı bir şekilde çok daha düşük seviyelerde tutmayı başarmıştır. Özellikle temel algoritmanın %23.2'lik birikimli sapma oranıyla tamamen kullanılamaz hale geldiği en zorlu senaryoda, AVL entegrasyonu bu oranı %1.5'e düşürerek sistemin çökmesini engellemiştir. Bu durum, önerilen AVL modülünün sadece artımlı bir iyileştirme sağlamakla kalmayıp, aynı zamanda VIO sistemini kısa süreli bir odometri tahmincisinden, büyük ve karmaşık ortamlarda sürekli doğrulukla seyrüsefer yapabilen, sağlam ve ölçeklenebilir bir lokalizasyon çözümüne dönüştüren temel bir bileşen olduğunu göstermektedir.



Şekil 5.7 VINS-Fusion (döngü kapatma aktif) – 100 m çembersel uçuş

Tablo 5.3 VINS-Fusion + AVL çalışma başarımının, referans VINS-Fusion algoritması ile *100m_iris_circle* uçuş sekansı üzerindeki karşılaştırma sonuçları

Metrik	**VINS-Fusion (Temel)**	**VINS-Fusion + LC**	**VINS-Fusion + AVL (Önerilen)**
RMSE (m)	236.90	270.09	**11.90**
Ortalama (m)	93.97	105.60	**9.71**
Medyan (m)	33.14	34.34	**8.94**
Maks. Hata (m)	1546.46	1562.38	**47.52**

Elde edilen sonuçlar, önerilen AVL entegrasyonunun küresel konum doğruluğu üzerinde önemli ölçüde iyileştirici bir etki yarattığını açıkça ortaya koymaktadır. Temel VINS-Fusion sisteminde 236.90 metre olarak hesaplanan konum hatası (RMSE), önerilen VINS-Fusion + AVL sistemi ile 11.90 metreye düşürülmüştür. Bu sonuç, yörünge doğruluğunda yaklaşık %95 oranında ciddi bir iyileşme sağlandığını göstermektedir.

Analiz kapsamında elde edilen en dikkat çekici bulgulardan biri, yaygın olarak kullanılan bir optimizasyon yaklaşımı olan Döngü Kapatma (Loop Closure, LC) yönteminin bu senaryoda beklenen performansı gösteremeyip, 270.09 metrelik RMSE değeriyle temel sistemden dahi daha yüksek bir hata oranı sergilemiştir. Bu bulgu, monoküler kamera sistemlerinde zorlu dış ortam koşullarının, döngü kapatma algoritmalarının yanlış eşleşmeler üretmesine ve dolayısıyla harita bütünlüğünün bozulmasına yol açabileceğini göstermektedir. Buna karşılık, AVL tabanlı sistemin mutlak coğrafi referans kullanması, bu tür hatalara karşı sistemi

büyük ölçüde bağışık hale getirmiş ve genel konumlama güvenilirliğini belirgin biçimde artırmıştır.

5.4 Veri Seti Sonuçları

Bu bölümde, tez kapsamında önerilen VINS-Fusion + AVL yaklaşımının performansı ve etkinliği, literatürde yaygın olarak kullanılan ve SLAM algoritmalarının değerlendirilmesinde zorlu senaryolar içermesiyle öne çıkan MARS-LVIG [52] veri seti üzerinde test edilmiştir. Öncelikle MARS-LVIG veri setinin temel özellikleri ayrıntılı olarak tanıtılmış, ardından elde edilen deneysel sonuçlar referans algoritmalarla karşılaştırmalı biçimde analiz edilmiştir.

5.4.1 MARS-LVIG Veri Seti

Algoritmanın objektif bir şekilde değerlendirilebilmesi amacıyla, bu çalışmada MARS-LVIG (Multi-sensor Aerial Robots SLAM for LiDAR-Visual-Inertial-GNSS fusion) veri seti kullanılmıştır. Bu veri seti, alçak irtifada (80–130 m) ve aşağı yönlü sensör konfigürasyonuyla toplanan verileri içermesiyle, mevcut VIO ve SLAM algoritmaları için özgün ve zorlu test koşulları sunmaktadır.

MARS-LVIG veri setinin ayırt edici özellikleri; IHA platformunda geniş ölçekli dış mekânlarda (yaklaşık 577,000 m²) toplanması, sensörlerin donanım seviyesinde senkronize edilmesi ve katı-hal LiDAR, global shutter RGB kamera, IMU ve ham GNSS verilerini birlikte sunmasıdır. Bu yapı, çoklu sensör füzyonuna dayalı yöntemlerin dayanıklılığını test etmek için kapsamlı bir altyapı sağlamaktadır.



Şekil 5.8 Rural town uçuş senaryosunun taranan bölge görseli

Bu tezde, veri setinden AMtown03 ve AMvalley03 adlı iki sekans seçilmiştir. Her iki sekans da yüksek dinamikler ve çevresel koşullar açısından algoritmalar için önemli zorluklar içermektedir.

Rural town senaryosu ve AMtown03 sekansı: Ermenistan'ın Ararat iline bağlı Urtsadzor kırsal bölgesi üzerinde, 80 m sabit irtifada ve 12 m/s hızda gerçekleştirilen bir uçuşa aittir. Toplam 5.109 km uzunluğundaki yörünge, 411,000 m²'lik bir alanı kapsamaktadır. Yüksek hız ve görsel olarak homojen kırsal doku, algoritmanın dinamik koşullardaki kararlılığını değerlendirmek için uygun bir test ortamı oluşturmaktadır.

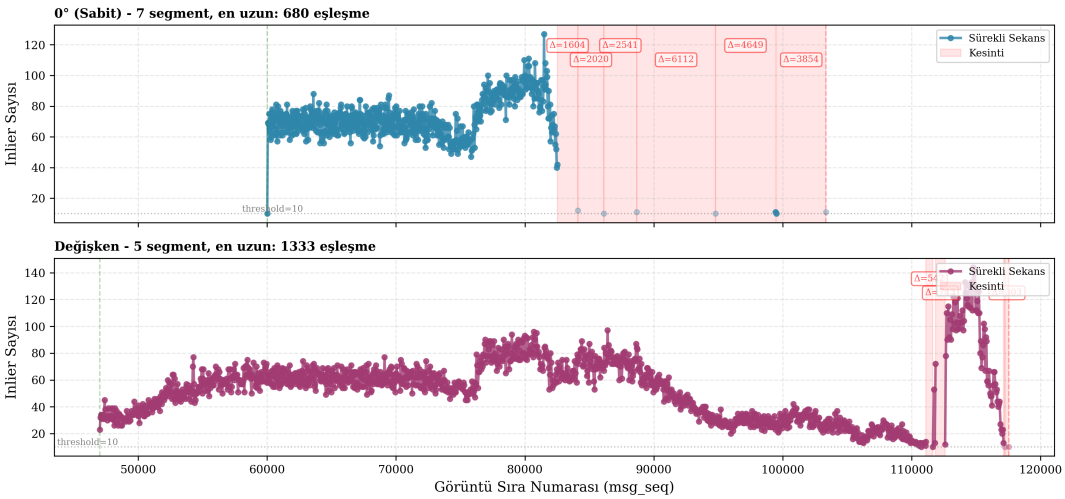


Şekil 5.9 Valley uçuş senaryosunun taranan bölge görseli

Valley senaryosu ve AMvalley03 sekansı: Ararat ilindeki Azizken Ormanı üzerinde, 130 m irtifada ve 12 m/s hızda kaydedilmiştir. 4.304 km'lik yörünge boyunca 577,000 m²'lik alan taranmıştır. Belirgin topografik değişimler içeren bu vadi yapısı, VIO algoritmalarının üç boyutlu ortam algılama ve irtifa değişimlerine adaptasyon yeteneklerini test etmek için uygun bir senaryo sunmaktadır.

5.4.2 Yöntem Değerlendirme ve Analiz

5.4.2.1 Görsel Dönmenin Öznitelik Eşleşme Doğruluğu Üzerindeki Etkisi



Şekil 5.10 Sabit ve değişken yaw stratejilerinin, AMvalley03 veri seti üzerindeki eşleştirme süreklilik profilleri

Şekil 5.10'te gösterilen süreklilik analizi, manyetometre tabanlı görsel döndürme stratejisinin SuperGlue eşleştirme performansı üzerindeki etkilerini ortaya koymaktadır. Sabit yaw ($0°$) ve değişken yaw stratejileri için elde edilen sonuçlar, SuperGlue'nun her iki yaklaşımda farklı performans özellikleri sergilediğini göstermektedir. Sabit yaw stratejisinde ortalama inlier sayısı 71.3 ± 13.1 olarak ölçülmüş olup, bu değer değişken yaw stratejisine göre yaklaşık $\%33.8$ daha yüksektir. Ancak UAV navigasyonu için kritik öneme sahip süreklilik açısından değerlendirildiğinde, değişken yaw stratejisi 1415 toplam eşleşme ile sabit yaw stratejisinin 687 eşleşmesine kıyasla **%106** daha fazla eşleştirme sağlamıştır.

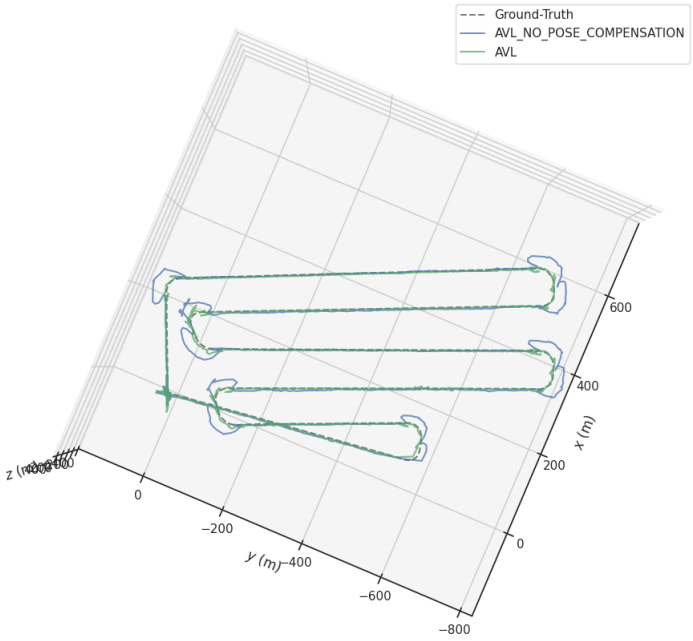
Şekil 5.10'in üst panelinde gösterilen sabit yaw stratejisi, 7 ayrı kesintisiz segment ile parçalı bir süreklilik profili sergilemektedir. En uzun segment 680 ardışık eşleştirme içermekte ve segmentler arası büyük boşluklar ($\Delta > 2000$ msg_seq) gözlemlenmektedir. Buna karşın, alt panelde gösterilen değişken yaw stratejisi yalnızca 5 segment ile daha kompakt bir profil sunmakta ve en uzun segment 1333 eşleştirme uzunluğuna ulaşmaktadır. Bu durum, değişken yaw stratejisinin uzun süreli eşlemelerde daha kesintisiz çalışabildiğini göstermektedir. Ortalama mesaj aralığı değerleri incelendiğinde, sabit yaw için 63.19 msg_seq olan bu değer, değişken yaw stratejisinde 49.88 msg_seq'e düşmektedir. Bu **%21.1**'lik fark, değişken yaw yaklaşımının daha yüksek eşleştirme frekansı sağladığını ortaya koymaktadır.

Bulgular, SuperGlue tabanlı eşleştirme performansının UAV yaw stratejisine bağlı olarak kritik bir kalite-süreklilik dengesi gösterdiğini ortaya koymaktadır. Sonuçlar, sabit yaw stratejisinin sınırlı süreklilik sunarken, değişken yaw yaklaşımının süreklilik ve sağlamlık açısından üstün olduğunu göstermektedir.

5.4.2.2 Yönelim Tabanlı Konum İyileştirme Performansı

Bu bölümde, Yönelim Tabanlı Konum İyileştirme metodolojisinin AVL algoritmasının konumlandırma doğruluğu üzerindeki etkisi, deneysel olarak analiz edilmiştir. Analizin temel amacı, önerilen yöntemin getirdiği katkıyı ortaya koymak ve etkinliğini doğrulamaktır.

Değerlendirme kapsamında iki farklı sistem ele alınmıştır. İlk olarak, *Temel AVL Sistemi (AVL_NO_POSE_COMPENSATION)*, yönelim tabanlı konum iyileştirme algoritmasını içermeyen ve standart AVL çıktısı olarak kullanılan referans sistemdir. İkinci olarak, *Önerilen Sistem (AVL)*, tez kapsamında geliştirilen ve yöntem bölümünde ayrıntılı olarak açıklanan konum düzeltme algoritmasının katkısını temsil eden sistemdir.



Şekil 5.11 Temel ve önerilen konum iyileştirmesinin AMtown03 veri seti üzerinde değerlendirmesi

Mutlak Yörünge Hatası analizi, geliştirilen konum iyileştirme yönteminin genel yörünge doğruluğuna olan katkısını en net şekilde ortaya koymaktadır. Her iki sistem için hesaplanan ATE istatistikleri, Tablo 5.4'da karşılaştırmalı olarak sunulmuştur.

Tablo 5.4 Temel ve önerilen yöntemlerin mutlak yörünge hatası karşılaştırması

Metrik	**Temel Sistem (m)**	**Önerilen Sistem (m)**	**İyileşme**
RMSE	**17.80**	**8.90**	**%50.0**
Ortalama (mean)	13.80	7.19	%47.9
Medyan (median)	14.35	6.90	%51.9
Std. Sapma (std)	11.23	5.24	-

Tablo 5.4'deki veriler incelendiğinde, önerilen konum iyileştirme yönteminin mutlak yörünge hatasını önemli ölçüde azalttığı görülmektedir. RMSE değeri, temel sistemde 17.80 metre iken, önerilen sistemde 8.90 metreye düşürülmüştür. Bu, geliştirilen metodoloji sayesinde toplam yörünge doğruluğunda **%50**'lik bir iyileşme sağlandığını göstermektedir.

Bu dikkate değer iyileşme, Yöntem bölümünde açıklanan ve hava aracının anlık duruş bilgilerini kullanarak konumunu sürekli olarak düzelten geri hesaplama mekanizmasının, zamanla biriken sürüklenme hatalarını başarılı bir şekilde

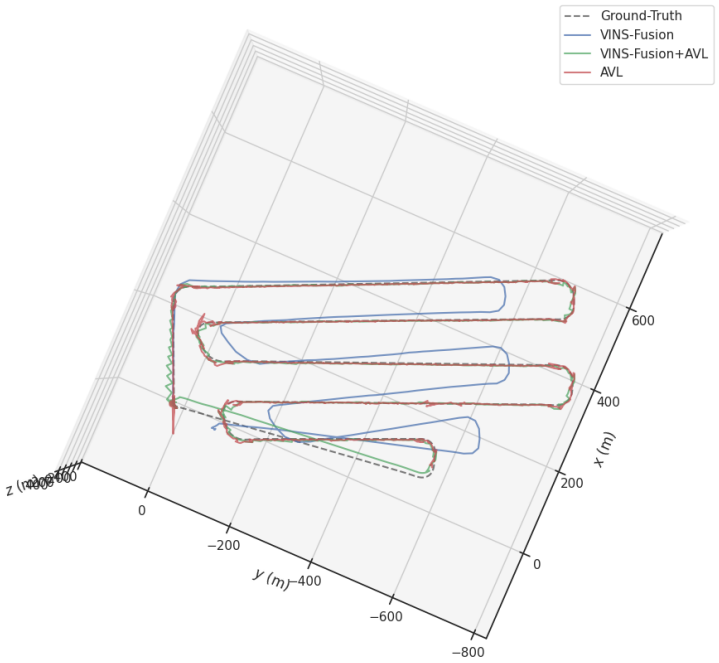
bastırdığının doğrudan bir kanıtıdır. Ortalama ve medyan hata değerlerindeki benzer oranlardaki düşüşler de bu bulguyu desteklemektedir.

Önerilen Yönelim Tabanlı Konum İyileştirme yönteminin, AVL sisteminin genel konum doğruluğunu önemli ölçüde artırdığını kesin olarak ortaya koyduğu gözlemlenmiştir. Mutlak yörünge hatasında elde edilen **%50**'lik RMSE iyileşmesi, geliştirilen metodolojinin teorik temellerinin pratikte ne denli etkili olduğunu doğrulamakta ve tezin temel hipotezlerinden birini başarıyla kanıtlamaktadır. Bu sonuç, yöntemin AVL tabanlı seyrüsefer sistemlerinin güvenilirliğini artırmada güçlü ve geçerli bir yaklaşım olduğunu göstermektedir.

5.4.2.3 Veri Seti Değerlendirmesi

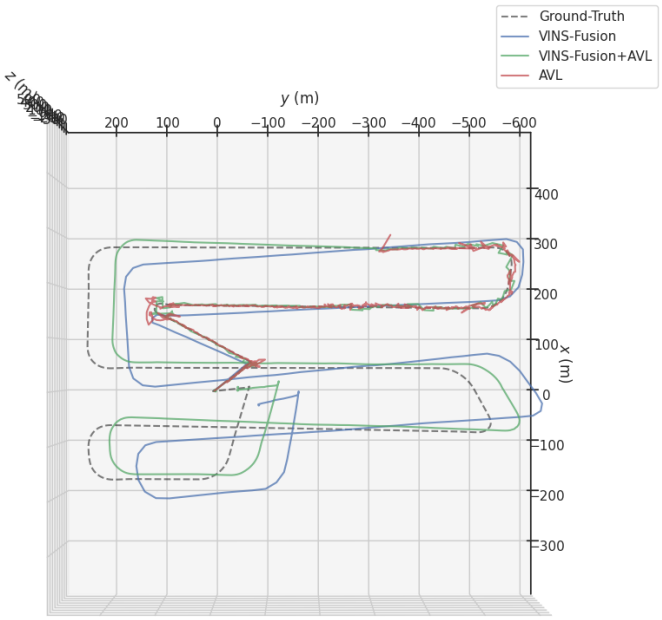
Geliştirilen çalışmanın yörünge tahmin doğruluğu, Mutlak Yörünge Hatası metriği kullanılarak, RMSE değeri üzerinden ölçülmüştür. Bu metrik, algoritmanın tahmin ettiği path ile veri setinde yer alan GNSS verisi arasındaki sapmayı ifade etmektedir.

Elde edilen sonuçların literatürdeki güncel çalışmalarla karşılaştırılabilmesi amacıyla, MARS-LVIG veri setinin referans makalesinde sunulan temel algoritmaların performansları esas alınmıştır. Bu yaklaşım, çalışmada geliştirilen yöntemin mevcut VIO ve LIO tabanlı çözümlerle kıyaslanabilirliğini sağlamak ve performans farklarını nicel olarak ortaya koymak amacıyla benimsenmiştir.



Şekil 5.12 AMtown03 uçuş senaryosunda VINS-Fusion, VINS-Fusion + AVL ve AVL çalışmalarından elde edilen trajektori sonuçları

Referans çalışmada, özellikle AMtown03 ve AMvalley03 sekanslarında yüksek uçuş hızının (yaklaşık 12 m/s) algoritmalar üzerindeki etkileri incelenmiştir. Bulgulara göre, bu yüksek hız koşullarında LIO-Livox, R3LIVE ve ORB-SLAM3 algoritmaları yörünge takibini başarıyla tamamlayamamıştır. Buna karşın, FAST-LIO2 algoritmasının her iki senaryoda da kararlı bir şekilde çalıştığı ve AMtown03 sekansında 2.93 m, AMvalley03 sekansında ise 8.21 m ATE (RMSE) değeri elde ettiği raporlanmıştır.



Şekil 5.13 AMvalley03 uçuş senaryosunda VINS-Fusion, VINS-Fusion + AVL ve AVL çalışmalarından elde edilen trajektori sonuçları

Tablo 5.5 Geliştirilen VINS-Fusion + AVL çalışmasının, referans VINS-Fusion ve FAST-LIO2 algoritması ile MARS-LVIG veri seti üzerindeki sonuçları

	FAST-LIO2		VINS-Fusion		VINS-Fusion + AVL	
Sequence	ATE (m)	Drift [%]	ATE (m)	Drift [%]	ATE (m)	Drift [%]
AMtown03	**2.93**	**0.057**	87.12	1.705	14.82	0.290
AMvalley03	**8.21**	**0.191**	73.47	1.707	45.12	1.048

Tablo 5.5' de ifade edilen sonuçlar, seçilen test sekanslarının yüksek dinamikler ve çevresel zorluklar nedeniyle mevcut algoritmalar için edge case niteliğinde olduğunu göstermektedir. Bu zorlu koşullar altında, tez kapsamında geliştirilen AVL modülünün, temel VINS-Fusion algoritmasının performansı üzerindeki etkisi incelendiğinde, önerilen yöntemin etkinliği açıkça ortaya konulmuştur. Özellikle AMtown03 sekansında, AVL entegrasyonu ile ATE değeri 87.12 metreden 14.82 metreye düşürülerek yaklaşık **%83** oranında kayda değer bir iyileşme sağlanmıştır.

Benzer şekilde, engebeli arazi yapısıyla öne çıkan AMvalley03 sekansında ise hata, **%38**'lik bir düşüşle 73.47 metreden 45.12 metreye gerilemiştir.

Bu çalışma, aynı zamanda yüksek maliyetli LiDAR ve düşük maliyetli monoküler kamera sensörlerini kullanan iki farklı yaklaşımın başarımını karşılaştırma imkanı sunmaktadır. Referans algoritma olan FAST-LIO2, LiDAR sensörünün sağladığı doğrudan derinlik bilgisi sayesinde her iki sekansta da en yüksek doğruluğu elde etmiştir. Ancak, önerilen AVL modülü ile güçlendirilmiş VIO sisteminin, AMtown03 sekansında LiDAR tabanlı sistemin doğruluğuna makul bir farkla yaklaşması kritik bir bulgudur. Bu durum, doğru algoritmik geliştirmelerle, tek bir kamera gibi maliyet-etkin bir sensör kullanarak dahi, zorlu dış mekan koşullarında LiDAR sistemlerinin sunduğu performansa yaklaşılabileceğini ve otonom sistemler için ölçeklenebilir çözümler üretilebileceğini göstermektedir.

6
SONUÇ

Sunulan çalışma, monoküler kamera donanımlı standart VINS-Fusion algoritmasının uzun süreli ve dinamik görevlerde, sürüklenme ve ölçek hatalarına karşı savunmasız olduğunu göstermiştir. Önerilen VINS-Fusion + AVL yaklaşımı ise bu zayıflıkları başarılı bir şekilde gidermektedir. AVL'den gelen ek bilgilerin füzyonu, sistemin hem anlık doğruluğunu hem de uzun vadeli kararlılığını önemli ölçüde iyileştirmiş, özellikle drift durumlarında konum kestiriminin referans yörüngeye sadık kalmasını sağlamıştır. Bu iyileşme, otonom seyrüsefer uygulamaları için kritik öneme sahip olan güvenilir ve tutarlı bir durum kestirimi yeteneği sunmaktadır.

Bu tez kapsamında geliştirilen AVL entegrasyonlu VINS-Fusion sistemi, GNSS'siz ortamlarda İHA navigasyonu için umut verici sonuçlar ortaya koymuştur. Ancak, sistemin performansını daha da iyileştirebilecek ve uygulama alanını genişletebilecek birtakım gelecek çalışma yönleri bulunmaktadır:

Yeni Nesil ve Çapraz Modalite Özellik Eşleştirme Algoritmaları: Mevcut sistemde kullanılan SuperGlue, yalnızca RGB modalitesinde ve benzer görüntüleme koşullarında etkili çalışmaktadır. Gelecek çalışmalarda iki farklı yönde iyileştirme yapılabilir: (i) Transformer tabanlı mimariler (LigthGlue [53], XoFTR [54]) büyük görüş açısı değişimleri ve zorlu aydınlatma koşullarında daha robust tek-modalite eşleştirme sağlayabilir. (ii) MINIMA [55] gibi çapraz modalite görüntü eşleştirme yöntemleri, farklı görüntüleme sistemleri arasında (RGB-termal, RGB-NIR, gündüz-gece vb.) eşleştirme yapabilme yeteneğine sahiptir. Bu sayede, gündüz-gece geçişlerinde termal kamera kullanımı, mevsimsel değişikliklerde farklı spektral bantlar arasında eşleştirme ve düşük ışık koşullarında NIR görüntülerin kullanılması mümkün hale gelebilir. MINIMA'nın üretken modeller kullanarak sentetik çok modlu eğitim verisi oluşturma yaklaşımı, sistemin farklı hava koşulları ve aydınlatma senaryolarına karşı generalizasyon yeteneğini önemli ölçüde artırabilir.

Fail-Safe ve Kurtarma Mekanizmaları: AVL'nin tamamen başarısız olduğu durumlarda (harita dışı bölgeler, eşleşme başarısızlığı), sistemin güvenli bir şekilde VIO moduna geri dönebilmesi ve görevine devam edebilmesi için fail-safe stratejileri tasarlanmalıdır. Bu stratejiler, AVL başarısızlığının otomatik tespiti ve sistematik mod değişikliği prosedürlerini içermelidir.

Hesaplama Yükü Optimizasyonu: SuperGlue gibi derin öğrenme tabanlı eşleştirme modellerinin hesaplama maliyeti, gerçek zamanlı uygulamalar için kritik bir sınırlayıcıdır. Model sıkıştırma teknikleri (pruning, quantization), hafif ağ mimarileri veya GPU tabanlı donanım ivmelendirme yöntemlerinin entegrasyonu ile sistemin gerçek zamanlı performansı iyileştirilebilir. Özellikle, NVIDIA Jetson serisi gibi embedded sistemler üzerinde optimizasyon çalışmaları yapılması, pratik uygulanabilirlik açısından önemlidir.

Bu öneriler, tez kapsamında geliştirilen sistemin potansiyel geliştirme alanlarını ve gelecek araştırma fırsatlarını ortaya koymaktadır. Her bir öneri, sistemin farklı bir yönünü geliştirmeyi hedeflemekte ve GNSS'siz İHA navigasyonunun pratikte daha güvenilir ve yaygın olarak kullanılabilir hale gelmesine katkı sağlayabilecek niteliktedir.

KAYNAKÇA

[1] J. R. Carr J. S. Sobek, "Digital Scene Matching Area Correlator (DSMAC)," *Image Processing for Missile Guidance*, T. F. Wiener, ed., International Society for Optics ve Photonics, c. 0238, SPIE, 1980, ss. 36–41. doi: 10.1117/12.959130. erişim adresi: https://doi.org/10.1117/12.959130.

[2] L. Matthies, C. Olson, G. Tharp S. Laubach, "Visual localization methods for Mars rovers using lander, rover, and descent imagery," *International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS), Tokyo, Japan*, Citeseer, 1997, ss. 413–418.

[3] A. I. Mourikis S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," *Proceedings 2007 IEEE international conference on robotics and automation*, IEEE, 2007, ss. 3565–3572.

[4] M. Abdollahi, S. H. Pourtakdoust, M. Y. Nooshabadi H. Pishkenari, "An improved Multi-State Constraint Kalman Filter for Visual-Inertial Odometry," *Journal of the Franklin Institute*, c. 361, no. 15, s. 107 130, Ekim 2024, issn: 0016-0032. doi: 10.1016/j.jfranklin.2024.107130. erişim adresi: http://dx.doi.org/10.1016/j.jfranklin.2024.107130.

[5] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang G. Huang, "Openvins: A research platform for visual-inertial estimation," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, ss. 4666–4672.

[6] G. Klein D. Murray, "Parallel tracking and mapping for small AR workspaces," *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, IEEE, 2007, ss. 225–234.

[7] R. Mur-Artal, J. M. M. Montiel J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE transactions on robotics*, c. 31, no. 5, ss. 1147–1163, 2015.

[8] D. Gálvez-López J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on robotics*, c. 28, no. 5, ss. 1188–1197, 2012.

[9] R. Mur-Artal J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, c. 33, no. 5, ss. 1255–1262, 2017.

[10] T. Qin, P. Li S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE transactions on robotics*, c. 34, no. 4, ss. 1004–1020, 2018.

[11] T. Qin, S. Cao, J. Pan S. Shen, "A general optimization-based framework for global pose estimation with multiple sensors," *arXiv preprint arXiv:1901.03642*, 2019.

[12] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE transactions on robotics*, c. 37, no. 6, ss. 1874–1890, 2021.

[13] S. Cao, X. Lu S. Shen, "GVINS: Tightly coupled GNSS–visual–inertial fusion for smooth and consistent state estimation," *IEEE Transactions on Robotics*, c. 38, no. 4, ss. 2004–2021, 2022.

[14] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[15] M.-M. Gurgu, J. P. Queralta T. Westerlund, *Vision-based GNSS-Free Localization for UAVs in the Wild*, 2022. arXiv: 2210.09727 [cs.RO]. erişim adresi: https://arxiv.org/abs/2210.09727.

[16] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer, 2022.

[17] A. Z. R. Hartley, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[18] D. A. Forsyth J. Ponce, *Computer vision: a modern approach*. prentice hall professional technical reference, 2002.

[19] C. L. G. Guoquan (Paul) Huang ve diğ., "Inertial Odometry for SLAM," içinde *SLAM Handbook: From Localization and Mapping to Spatial Intelligence*, A. K. Luca Carlone ve diğ., ed., Accessed: March, 2025, Cambridge, UK: Cambridge University Press, 2025, böl. 13. erişim adresi: https://github.com/SLAM-Handbook-contributors/slam-handbook-public-release/blob/main/main.pdf.

[20] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[21] P. S. Maybeck, *Stochastic Models, Estimation, and Control* (Mathematics in Science and Engineering). New York: Academic Press, 1979, c. 1, isbn: 978-0124808018.

[22] S. J. Julier J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," *Proceedings of SPIE - The International Society for Optical Engineering*, A. '97, ed., c. 3068, SPIE, 1997, ss. 182–193. doi: 10.1117/12.280797.

[23] F. R. Kschischang, B. J. Frey H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on information theory*, c. 47, no. 2, ss. 498–519, 2002.

[24] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Magazine*, c. 21, no. 1, ss. 28–41, 2004.

[25] R. Chatila J. Laumond, "Position referencing and consistent world modeling for mobile robots," *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, IEEE, c. 2, 1985, ss. 138–145.

[26] F. Lu E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous robots*, c. 4, ss. 333–349, 1997.

[27] F. Dellaert M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *The International Journal of Robotics Research*, c. 25, no. 12, ss. 1181–1203, 2006.

[28] S. Thrun M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures," *The International Journal of Robotics Research*, c. 25, no. 5-6, ss. 403–429, 2006.

[29] C. Taylor J. Gross, "Factor graphs for navigation applications: A tutorial," *NAVIGATION: Journal of the Institute of Navigation*, c. 71, no. 3, 2024.

[30] B. J. Frey, F. R. Kschischang, H.-A. Loeliger N. Wiberg, "Factor graphs and algorithms," *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, Citeseer, c. 35, 1997, ss. 666–680.

[31] S. Thrun, W. Burgard D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.

[32] J. Nocedal S. J. Wright, *Numerical optimization*. Springer, 1999.

[33] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," *Georgia Institute of Technology, Tech. Rep*, c. 2, no. 4, 2012.

[34] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige W. Burgard, "g 2 o: A general framework for graph optimization," *2011 IEEE international conference on robotics and automation*, IEEE, 2011, ss. 3607–3613.

[35] T. A. Davis, *Direct methods for sparse linear systems*. SIAM, 2006.

[36] S. Agarwal, K. Mierle ve diğ., "Ceres solver: Tutorial & reference," *Google Inc*, c. 2, no. 72, s. 8, 2012.

[37] G. Guennebaud, B. Jacob ve diğ., *Eigen: A C++ template library for linear algebra*, 2023. erişim adresi: `http://eigen.tuxfamily.org`.

[38] T. A. Davis, "Algorithm 1000: SuiteSparse: GraphBLAS: Graph algorithms in the language of sparse linear algebra," *ACM Transactions on Mathematical Software (TOMS)*, c. 45, no. 4, ss. 1–25, 2019.

[39] W. Wen, T. Pfeifer, X. Bai L.-T. Hsu, "Factor graph optimization for GNSS/INS integration: A comparison with the extended Kalman filter," *NAVIGATION: Journal of the Institute of Navigation*, c. 68, no. 2, ss. 315–331, 2021.

[40] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *The International Journal of Robotics Research*, c. 31, no. 2, ss. 216–235, 2012.

[41] K. Eckenhoff, P. Geneva G. Huang, "Closed-form preintegration methods for graph-based visual–inertial navigation," *The International Journal of Robotics Research*, c. 38, no. 5, ss. 563–586, 2019.

[42] T. Lupton S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Transactions on Robotics*, c. 28, no. 1, ss. 61–76, 2011.

[43] L. Carlone, Z. Kira, C. Beall, V. Indelman F. Dellaert, "Eliminating conditionally independent sets in factor graphs: A unifying perspective based on smart factors," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, ss. 4290–4297.

[44] F. Dellaert, M. Kaess ve diğ., "Factor graphs for robot perception," *Foundations and Trends® in Robotics*, c. 6, no. 1-2, ss. 1–139, 2017.

[45] D. Fourie, A. T. Espinoza, M. Kaess J. Leonard, "Characterizing marginalization and incremental operations on the Bayes tree," *International Workshop on the Algorithmic Foundations of Robotics*, Springer, 2020, ss. 227–242.

[46] T. A. Davis, J. R. Gilbert, S. I. Larimore E. G. Ng, "Algorithm 836: COLAMD, a column approximate minimum degree ordering algorithm," *ACM Transactions on Mathematical Software (TOMS)*, c. 30, no. 3, ss. 377–380, 2004.

[47] T. Qin, J. Pan, S. Cao S. Shen, *A General Optimization-based Framework for Local Odometry Estimation with Multiple Sensors*, 2019. eprint: arXiv:1901.03638.

[48] T. Qin S. Shen, "Online Temporal Calibration for Monocular Visual-Inertial Systems," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, ss. 3662–3669.

[49] M. Schleiss, F. Rouatbi D. Cremers, *VPAIR – Aerial Visual Place Recognition and Localization in Large-scale Outdoor Environments*, 2022. arXiv: 2205.11567 [cs.CV]. erişim adresi: https://arxiv.org/abs/2205.11567.

[50] P.-E. Sarlin, D. DeTone, T. Malisiewicz A. Rabinovich, *SuperGlue: Learning Feature Matching with Graph Neural Networks*, 2020. arXiv: 1911.11763 [cs.CV]. erişim adresi: https://arxiv.org/abs/1911.11763.

[51] M. FISCHLER AND, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, c. 24, no. 6, ss. 381–395, 1981.

[52] H. Li ve diğ., "MARS-LVIG dataset: A multi-sensor aerial robots SLAM dataset for LiDAR-visual-inertial-GNSS fusion," *The International Journal of Robotics Research*, c. 43, no. 8, ss. 1114–1127, 2024.

[53] P. Lindenberger, P.-E. Sarlin M. Pollefeys, "Lightglue: Local feature matching at light speed," *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, ss. 17 627–17 638.

[54] Ö. Tuzcuoğlu, A. Köksal, B. Sofu, S. Kalkan A. A. Alatan, "Xoftr: Cross-modal feature matching transformer," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, ss. 4275–4286.

[55] J. Ren, X. Jiang, Z. Li, D. Liang, X. Zhou X. Bai, "Minima: Modality invariant image matching," *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, ss. 23 059–23 068.

TEZDEN ÜRETİLMİŞ YAYINLAR

Makale

1. F. Başatemur and E. Uslu, *Outdoor Application of Visual-Inertial Odometry and SLAM Techniques*, in *Proceedings of the YTU Graduate School of Science and Engineering GradColloquium 2024 - Artificial Intelligence*, İstanbul, Türkiye, 2024, pp. 51–52.