

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE
ENGINEERING AND TECHNOLOGY

**ACTIVE SLAM WITH INFORMATIVE PATH PLANNING FOR
HETEROGENEOUS ROBOT TEAMS**



Ph.D. THESIS

Mehmet Caner AKAY

Department of Control and Automation Engineering

Control and Automation Engineering Programme

OCTOBER 2020

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE
ENGINEERING AND TECHNOLOGY

**ACTIVE SLAM WITH INFORMATIVE PATH PLANNING FOR
HETEROGENEOUS ROBOT TEAMS**



Ph.D. THESIS

Mehmet Caner AKAY
(504122104)

Department of Control and Automation Engineering

Control and Automation Engineering Programme

Thesis Advisor: Prof. Dr. Hakan TEMELTAŞ

OCTOBER 2020

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**HETEROJEN ROBOT TAKIMLARI İÇİN BİLGİLENDİRİCİ YOL
PLANLAMALI AKTİF EZKH**

DOKTORA TEZİ

**Mehmet Caner AKAY
(504122104)**

Kontrol ve Otomasyon Mühendisliği Bölümü

Kontrol ve Otomasyon Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Hakan TEMELTAŞ

EKİM 2020

Mehmet Caner AKAY, a Ph.D. student of ITU Graduate School of Science Engineering and Technology student ID 504122104 successfully defended the thesis/dissertation entitled “ACTIVE SLAM WITH INFORMATIVE PATH PLANNING FOR HETEROGENEOUS ROBOT TEAM”, which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Hakan TEMELTAŞ**
Istanbul Technical University

Jury Members : **Prof. Dr. Osman Nuri UÇAN**
Altınbaş University

Assoc. Prof. Dr. Gülay ÖKE GÜNEL
Istanbul Technical University

Assoc. Prof. Dr. Volkan SEZER
Istanbul Technical University

Assoc. Prof. Dr. Ahmet YAZICI
Eskişehir Osmangazi University

Date of Submission : 26 August 2020

Date of Defense : 15 October 2020





To my dear family,



FOREWORD

There are many people I would like to acknowledge as being a part of this dissertation. First of all, I would like to express my deepest gratitude to my advisor Prof. Dr. Hakan Temeltaş, for his support and supervision. His guidance during the development of this thesis was invaluable to me. Also, I would like to thank him for providing the opportunities to access the ITU Robotics Laboratory to me. I also wish to express my gratitude to the academic and technical staff in the Department of Control Engineering for their support.

I would also like to express my sincere gratitude to my thesis progress members Prof. Dr. Osman Nuri Uçan and Assoc. Prof. Dr. Gülay Öke Günel, with my thesis defence jury members Assoc. Prof. Dr. Ahmet Yazıcı and Assoc. Prof. Dr. Volkan Sezer for their valuable contributions, comments and guidance to my studies. I also wish to state my gratitude to my previous advisor at Defense Technologies MSc. Program, Gökhan İnalhan, who influenced me to start my PhD. Without his referral, I couldn't manage to find the right academic career path.

I would also like to thank all my labmates, especially Garen Haddeler, who helped me to overcome several problems during my thesis. I am confident that he will keep accomplishing many things.

I also wish to express my gratitude to my managers at ALTINAY. I would like to thank the Chairman of the Board, Hakan Altınay, General Manager, M. Emin İlkmen and Deputy General Manager, Z. Burak Mercan, for their belief in completing my PhD, support and the opportunities they provided. In addition, I would like to thank my colleague Duruhan Özçelik for his help during my thesis writing progress.

I would like to thank my dearest friend Çağatay K. Aktaş for his support and being there for me in difficult times. Same goes for Buğra Çonguloğlu, Coşkun Demirtaş, Erdinç Özkıran, Gözde Konca, Hazar Mühürücü, Melike Dilek, Ömer Emeksiz, Sina Küçükosman, Şefikcan Biricik and Utku Uzun who are the ones that I can spend the most amazing times together. Also, I would like to thank my friends, Çağrı Öztürk, Gökalp Yanıkçı, Gökhan Keskin, Gürcan Eralp, İbrahim B. Yeşiloğlu, H. İhsan Altundaş, Dr. Önder Umut and Serkan Köse, whom I know them from FENERBAHÇE S.C. tribune. Without them and FENERBAHÇE, it would be hard to relax during tough times. I couldn't have done it without my friends. And most importantly, I would like to thank Meltem Ertekin for her toleration at my stressful times and optimistic approaches to save me from desperate situations. She is my other half, source of happiness and has a special place in my heart.

My final thanks are reserved for my family and relatives. I would like to express my heartfelt gratefulness for their love, support and belief that I will succeed. I am fortunate to have S. İrem Akay as my sister, and I would like to thank her for all the suggestions she gave on balancing my academic and social life and as being a bridge between my parents and me to reach a compromise. I would like to thank my mother, Ş. Gülşen Akay, for her continual love and belief shown by her saying, "You can

handle it.”. It wouldn’t be possible to complete my thesis without her hopeful attitude. I would also like to thank my father, İ. Taner Akay as being my idol and guide during my whole life. He always encouraged me to improve myself and achieve my goals. I am so grateful to have my family and to honour them with my achievements.

This study was supported by the TUBITAK 116E178 project.

October 2020

Mehmet Caner AKAY
(M. Sc.)



TABLE OF CONTENTS

	<u>Page</u>
FOREWORD	ix
TABLE OF CONTENTS	xi
ABBREVIATIONS	xiii
SYMBOLS	xv
LIST OF TABLES	xix
LIST OF FIGURES	xxi
SUMMARY	xxiii
ÖZET	xxv
1. INTRODUCTION	1
1.1 Purpose of Thesis	2
1.2 Literature Review	3
1.3 Contributions	8
1.4 Thesis Outline	8
2. SLAM METHOD	11
2.1 EKF Localization	11
2.2 Lidar Odometry and Mapping.....	12
2.3 OctoMap.....	15
2.4 Height Mapping.....	17
3. CONSTRUCTING COMMON MAP	21
3.1 Layering Method	22
3.1.1 Layered Map of UGV	23
3.1.2 Layered Map of UAV	23
3.2 Similarity Metrics.....	24
4. INFORMATIVE PATH PLANNING	29
4.1 Relative Entropy.....	29
4.2 Information Seeking Optimal Control.....	31
5. CASE STUDIES	37
5.1 Experimental Studies.....	37
5.1.1 Hardware Setup.....	37
5.1.1.1 Integrated experimental setup	38
5.1.2 Software Setup	39
5.1.3 Experiment Area	40
5.1.4 Scenario.....	41
5.1.5 Results.....	43
5.1.5.1 Height Value RMSE	43
5.1.5.2 Visual Outputs.....	45
5.2 Simulation Studies.....	51
5.2.1 Simulation Setup	51
5.2.1.1 Robot Operating System	52
5.2.1.2 Gazebo Simulator.....	55
5.2.2 Scenario.....	56
5.2.3 Results.....	57

5.2.3.1 Visual Outputs.....	59
5.2.3.2 Performance Outputs.....	62
6. CONCLUSION.....	67
6.1 Practical Applications of This Study.....	68
6.2 Future Work.....	69
REFERENCES.....	71
APPENDICES.....	77
APPENDIX A.....	78
APPENDIX B.....	84
APPENDIX C.....	85
APPENDIX D.....	86
CURRICULUM VITAE.....	90



ABBREVIATIONS

2D	: Two Dimension
2.5D	: Two and a Half Dimension
3D	: Three Dimension
BS	: Belief Space
CPU	: Central Processing Unit
D-RTK	: DJI Real-time Kinematic
EKF	: Extended Kalman Filter
GPS	: Global Positioning System
HeRT	: Heterogeneous Robot Team
HoRT	: Homogeneous Robot Team
IMU	: Inertial Measurement Unit
JD	: Jeffrey Divergence
JeD	: Jensen Divergence
JeSD	: Jensen-Shannon Divergence
KD	: K Divergence
KLD	: Kullback-Liebler Divergence
Lidar	: Light Detection and Ranging
LOAM	: Lidar Odometry and Mapping
QR	: Quick Response
RAM	: Random-Access Memory
RMSE	: Root Mean Square Error
ROS	: Robot Operating System
SE	: Shannon Entropy
SLAM	: Simultaneous Localization and Mapping
TD	: Topsoe Divergence
UAV	: Unmanned Aerial Vehicle
UGV	: Unmanned Aerial Vehicle
VSLUT	: Volumetric Spaces Left Under the Threshold
VSE	: Volumetric Spaces of the Environment



SYMBOLS

C	: Capacitance
k	: States
X_k	: States
\bar{X}_k	: Predicted states of the robot
\bar{P}_k	: Predicted covariance
f	: State function
F_k	: Jacobian of f
R_k	: Process noise
K_k	: Kalman gain
h	: Observation function
H_k	: Jacobian of h
m_k	: Measurements
I	: Identity matrix
x_k, y_k, z_k	: Positions of the robot in three axis
φ, θ, ψ	: Orientation in Euler angles
κ	: Surface roughness
G	: Set of points obtained during a scan
$C_{n,i}$: Coordinates of points
Γ_n, Δ_n	: Edge points and surface points, respectively
d_Γ, d_Δ	: Distance from edge points and surface points, respectively
S_n	: Map obtained by each point
T_σ^L	: Position and localization difference
p_x, p_y, p_z	: Position difference in three coordinates
$\theta_x, \theta_y, \theta_z$: Orientation difference in Euler angles
t	: Moment at the investigation
t_i	: Initial time
t_σ	: Duration of the scan in one rotation
R	: Rotation matrix
J	: Jacobian of nonlinear distance
λ	: Factor provided by Lavenberg-Marquardt method
M_{LOAM_t}	: Map obtained by LOAM method
V_i	: Voxel
$P(V_i m_t)$: The joint probability of voxel's occupancy state
$l(V_i m_{1:t})$: The joint probability of voxel's occupancy state, log-odds presentation
l_{UB_o}	: Upper bound of OctoMap
l_{LB_o}	: Lower bound of OctoMap
$H_{0:t}$: Height Map
g_i	: Grid
$O_{0:t}$: Occupancy probability of each grid
$\mu_{0:t}$: Height estimation mean value
$\sigma_{0:t}^2$: Variance of height

z_t	: Height value
$P(g_i m_t)$: Joint probability of grid's occupancy state
$l(g_i m_{1:t})$: Joint probability of grid's occupancy state, log-odds presentation
h_n	: Height of the layer
u	: Position and orientation difference
T	: Translation
\tilde{u}	: Best position and orientation difference
$H_{bd}(u)$: Entropy between draft map and base map
$d_{bd}(u)_{JD}$: Divergence between draft map and base map
$G_m(z), A_m(z)$: Obtained ground and aerial height map, respectively
$P(w_s)$: Occupancy probability of volumetric space in the whole map
$P(o_s)$: Occupancy probability of volumetric space in the observed map
d_{KL}	: Kullback-Leibler Divergence
M_w	: Whole map
M_o	: Observed map
w_s	: Environmental states in volumetric space
o_s	: Observed states in volumetric space
$I(w_s, o_s)$: Mutual information
$H(w_s)$: The entropy of environmental states
$H(w_s o_s)$: Conditional entropy of observed states over environmental states
$P(w_s, o_s)$: Joint probability of environmental states and observed states
$u^r(t)$: Robot information seeking control input at time t
O	: Objective function
$p_{c_{ugv}}, p_{c_{uav}}$: Target points of UGV and UAV, respectively
x_{pc}, y_{pc}, z_{pc}	: Points of interest in three axis
ω_i	: Weight value
V_k	: Lyapunov candidate function
I_{th}	: Information threshold
$D_{tolerance}$: Tolerance distance
u_k	: Robot control input at time interval k
Δu_e	: Eastside wheel motion difference
Δu_w	: Westside wheel motion difference
v_k	: Velocity vector
ϑ_k	: Sensor measurement noise
S_k	: Observation uncertainty
Q_{ϑ_k}	: The covariance of sensor measurement noise
μ	: Map mean value
Σ	: Covariance of map
O_{DWA}	: The objective function of the Dynamic Window Approach
S_{V_a}	: Set of acceptable velocities
S_{V_d}	: Set of DWA velocities
σ	: Heading weight
φ	: Distance weight
v	: Velocity weight
\dot{x}	: Translational velocity
$\dot{\theta}$: Rotational velocity
σ_{nn}^2	: Variances of sensors
a	: Equatorial radius of the earth

c : Polar radius of the earth





LIST OF TABLES

	<u>Page</u>
Table 5.1 : Experimental study equipment	37
Table 5.2 : Data rates of the hardware	38
Table 5.3 : Number of iterations to calculate the best position and orientation with and without the layering method for different similarity metrics	44
Table 5.4 : ROS packages, parameters, their values and the reasons for selected values.....	53
Table 5.5 : Parameters were used in cases of the study.	58
Table 5.6 : Exploration performances of cases.	62
Table D.1 : MATRICE 600PRO main features	86
Table D.2 : HUSKY A200 features.	87
Table D.3 : D-RTK GPS features.....	87
Table D.4 : Velodyne VLP-16 specifications.	8627
Table D.5 : Manifold technical specifications.	88



LIST OF FIGURES

	<u>Page</u>
Figure 2.1 : Edge (p and r) and surface (s,v and l) points.....	13
Figure 2.2 : Different types of mapping results.	16
Figure 2.3 : Lines with different colours show the representative laser beams for one channel.	18
Figure 2.4 : Height map 2.5D X-Y plane projection.	18
Figure 2.5 : Height map 3D illustration.	19
Figure 3.1 : Working principle of common map construction.....	21
Figure 3.2 : A representative figure of the heights of the layers from the experimental study used in this dissertation.....	22
Figure 3.3 : UGV layered local height maps.	23
Figure 3.4 : UAV layered local height maps.	24
Figure 4.1 : Informative path planning block diagram.	29
Figure 4.2 : Venn scheme representation of the KLD.	30
Figure 4.3 : Representation of conditional entropy of states.	31
Figure 4.4 : Representative illustration of volumetric spaces.....	35
Figure 4.5 : Flow diagram of informative path planning in this dissertation.....	36
Figure 5.1 : Experimental setup principle of operation.	38
Figure 5.2 : UGV experimental setup-outside	39
Figure 5.3 : Experimental setup physical and functional block diagram.....	40
Figure 5.4 : Experiment area photo was taken by UAV.	41
Figure 5.5 : Chart of similarity metric versus computation time and RMSE.	44
Figure 5.6 : Box plot of the similarity metrics according to RMSE values.....	45
Figure 5.7 : Obtained height maps of the layers.	46
Figure 5.8 : Obtained height maps of the layers.	47
Figure 5.9 : Obtained common height map of the environment.	48
Figure 5.10 : Obtained OctoMap of the environment.....	49
Figure 5.11 : Common point cloud map obtained with Jensen Divergence similarity metric output transform in the form of different camera line of sight. ...	50
Figure 5.12 : Trajectory input and odometry output of the experiments.	50
Figure 5.13 : Simulation setup functional block diagram.	51
Figure 5.14 : Gazebo simulation scene for this dissertation.	55
Figure 5.15 : Advantages of employing HeRT.	58
Figure 5.16 : Obtained height map of the environment by HeRT.	59
Figure 5.17 : Obtained OctoMaps of the environment from (a) to (f), with regard to cases 1 to 6, respectively.	60
Figure 5.18 : Lack of information (red circles) and distortions of visual outputs (yellow circles).	61
Figure 5.19 : Total movement of the team and volumetric spaces left under the threshold according to the total number of volumetric spaces, case by case.	62
Figure 5.20 : Evolution of collected total information, duration of the mission and the memory usage of the obtained map, case by case.	63

Figure 5.21 : The entropy evolution during the exploration mission, case by case..	65
Figure A.1 : Non-holonomic motion with two-wheel encoders trajectory representation.	78
Figure A.2 : Constant velocity model representation.....	80
Figure A.3 : An illustrated form of sensor types and examples of products.....	80
Figure A.4 : UGV obstacle avoidance features.....	82
Figure D.1 : DJI MATRICE 600PRO.....	86



ACTIVE SLAM WITH INFORMATIVE PATH PLANNING FOR HETEROGENEOUS ROBOT TEAMS

SUMMARY

Recently, heterogeneous teams consisting of unmanned ground vehicles and unmanned aerial vehicles are being used for different types of missions such as surveillance, tracking, and exploration, etc. Exploration missions with heterogeneous robot teams should acquire a common map for understanding the surroundings better. The unique approach presented in this dissertation with cooperative use of agents provides a well-detailed observation over the environment where challenging details and complex structures are involved. Also, the presented method is suitable for real-time applications and autonomous path planning for exploration.

Lidar Odometry and Mapping with various similarity metrics such as Shannon Entropy, Kullback-Liebler Divergence, Jeffrey Divergence, K Divergence, Topsoe Divergence, Jensen-Shannon Divergence and Jensen Divergence are used to construct a common height map of the environment. Furthermore, the given layering method that provides more accuracy and a better understanding of the common map. All of the given similarity metrics are compared, and the advantage of utilizing the layering method is shown. The best similarity metric for constructing a heterogeneous robot team common map of the experimental area was obtained by using the Jensen Divergence similarity metric and layering method.

Moreover, Extended Kalman Filter localization and OctoMap techniques are utilized to create an adaptive simultaneous localization and mapping infrastructure for informative path planning. Optimal parameter tuning for the specified simulation environment provides adjustable memory allocation and exploration performance, such as; duration, collected information and effort.

The information seeking controller obtained with the use of relative entropy ensures exploration of the given area to minimize the uncertainty between observed states and environmental states. Robots move to the volumetric spaces' center under given rules and collect measurements by proprioceptive and exteroceptive sensors. With the use of heterogeneous robot teams, the measurements collected by the Lidar provide an advantage in perceiving complex details that can not be done by homogeneous robot teams.

Constructing common map part of the theoretical approaches in this thesis are experimentally validated. In addition, the complete demonstration of this dissertation is done with six different cases by simulation studies. The theoretical background of active simultaneous localization and mapping with informative path planning for heterogeneous robot teams are validated, and the advantages of this study are remarked.



HETEROJEN ROBOT TAKIMLARI İÇİN BİLGİLENDİRİCİ YOL PLANLAMALI AKTİF EZKH

ÖZET

İnsansız Hava Aracı'nı (İHA'yı) ve İnsansız Kara Aracı'nı (İKA'yı) bünyesinde bulunduran heterojen yapıları robot takımları, günümüzde gözetleme, takip, keşif, vb. farklı görevlerde kullanılmaktadır. Çevrenin haritalanmasını gerektiren keşif görevlerinde, heterojen robot takımlarının ortamı daha iyi anlayabilmesi adına, ortak bir haritaya ihtiyaç duyulmaktadır. Bu doğrultuda özel yaklaşımlarla, Lidar Odometre ve Haritalama (LOH) ile zorlayıcı yapıların bulunduğu ortamda, araçların kooperatif bir şekilde benzerlik metriklerini kullanarak ortak harita çıkarması sağlanmaktadır. Bunun yanı sıra, sınırları belirli bir alanın, heterojen robot takımları ile keşfini sağlamak adına sürekli olarak toplanan bilgiyi arttırıcı kontrolcü tasarımı kullanılmaktadır.

Farklı tipte hareket denklemlerine ya da dinamik modellere ve/veya farklı sensör yapılarına sahip robotlardan oluşan robot takımlara heterojen yapıları robot takımları denmektedir. Diğer taraftan robot takımlarının eş zamanlı konumlama ve haritalama problemi ile bu takımdaki robotların yol planlamalarının eş zamanlı gerçekleşmesi ise Aktif eş zamanlı konumlama ve haritalama (EZKH) problemi olarak adlandırılmaktadır. Buradaki eş zamanlı gerçekleştirilmedeki amaç otonom robot araçları için planlanan yolların aynı zamanda EZKH'deki belirsizliği de minimize edecek şekilde gerçekleştirilmesidir. Diğer bir deyişle otonom robot araçları için bilgilendirici yol planlarının oluşturulmasıdır. Bu çalışmanın temel amacı heterojen yapıları robot grupları için bilgilendirici yol planlamaya dayalı bir Aktif-EZKH sistemi tasarlamaktır.

Robot takımlarının farklı dinamik ve sensörlere sahip olması diğer bir deyişle heterojen yapıda olmaları, bu robot takımlarına avantajlar getirmektedir. Örneğin; hava robotları hızlı hareket edebilir, kara robotları daha ağır faydalı yükler taşıyabilir ve hedef nokta ile doğrudan etkileşime girebilirler. Karma bir araçlı bir yapı içerisinde yer alan İHA ile İKA oluşan bir robot grubu keşif, arama veya güvenlik amaçlı sınırları belirlene bir bölge içinde iş birliği yaparak ortam içindeki görevlerini insandan bağımsız bir şekilde otonom olarak gerçekleyebilir. Burada İHA ve İKA'ların birbirleri ile yer istasyonu aracılığı ile veri paylaşımında olduğu varsayılmaktadır. Komşuluk alanları içerisinde haberleşme ile harita paylaşımı ya da araç durum vektörü paylaşımı yapabilen robot birimleri kooperatif robotlar olarak gösterilmektedir. Bu çalışmada dış ortam sensörü olarak 360° ortam taraması yapabilen ve saniyede 300.000 adet noktanın mesafesini ölçebilen 3B LIDAR sistemi kullanılmıştır. Yüksek çözünürlüklü ölçüm avantajının yanı sıra bu kadar büyük miktardaki veriden optimum miktarda ve hızlı bir şekilde anlamlı veri üretip bunları robot konumlama, planlama ve koordinasyonunda kullanım da ayrı bir zorluk ortaya koymaktadır.

Temel olarak, hareketli olan bir araçtan elde edilen nokta bulutunun coğrafik olarak yerleştirilmesi gerekmektedir. Bu işlem sadece Lidar sensörü kullanılarak da; farklı sensörlerin verilerinin ortak bir şekilde kullanılması aracılığıyla da yapılabilir. Sadece

Lidar ile toplanmış verilerin işlenerek nokta bulutunun coğrafik olarak yerleştirilmesi ve gözlem sırasında sensörün hareketinin elde edilmesi, bu çalışmada LOH ile sağlanmaktadır. Bu sayede; GPS ve IMU olmaksızın EZKH yapılabilir. Buna ek olarak, sensörlere binen gürültülerden dolayı oluşabilecek kaymalar ve yanlış veri elde edilmesi engellenebilmektedir. Buna karşılık, GPS, enkoder ve IMU verileri ile Lidar verileri birleştirilerek Genişletilmiş Kalman Filtresi (GKF) konumlaması da sağlanabilmektedir. Burada sensör verilerinin olasılıksal yaklaşımlarla işlenmesi ile robotun konumu elde edilmektedir ve bu konum ile Lidar verilerinin coğrafik yerleştirme yapılması sonucunda da belirli bir orijine sabitlenmiş nokta bulutu çıktısı alınmaktadır. Sonrasında da bu nokta bulutu ile istenilen yöntem ile elde edilen odometre ve nokta bulutu verisi farklı haritalama yöntemleri kullanılarak ayarlanabilir özel görsel çıktılar sağlanabilmektedir. Bunlardan biri, sekizli ağaç yapıları kullanılarak elde edilen OctoMap olmaktadır. OctoMap yöntemi, tez çalışmasında kullanılmasının temel sebepleri olan, çözünürlük ayarlaması, doluluk olasılığı üst ve alt sınırları belirlenmesi ve 3B olarak sağlanabilmesi açısından faydalı bir araç olmaktadır. Bu yöntem ile, ortamın uyarlanabilir şekilde, ortamın 3B haritasının çıkarılması sağlanmaktadır.

Lidar sensörlerinin havadan alınan nokta bulutları ile karadan alınan nokta bulutları farklı geometrik özellikler taşımaktadır. Ancak, hava ve kara Lidar görüntülemesinin birbirlerini tamamlaması bakımından oldukça büyük avantajları da mevcuttur. Hava aracı ve kara aracı tarafından yapılan ve birilerinin göremedikleri bölgelerin görüntülenebilmesi sağlanabilmektedir. Bu avantajı kullanabilmek adına farklı açılardan lokal olarak görüntülenen ortamın ortak bir haritada birleştirilmesi gerekmektedir. Harita birleştirme adımını gerçekleştirmek adına her iki robotun elde ettiği verilerden ortak olanını belirlemek gerekmektedir. Kuş bakışı veya yatay olması fark etmeksizin bir nesnenin yere göre yüksekliği; hem havadan hem karadan yapılan gözlemlerde sensörlerin görüş açısı sınırları içerisinde aynı olacaktır. Bu doğrultuda, yükseklik verileri üzerinden benzerlik metrikleri kullanılarak haritaların birleştirilmesi sağlanabilmektedir. Bu tez çalışmasında, İHA ve İKA tarafından elde edilen nokta bulutu ızgara haritasına benzer bir yapıda olan yükselti haritaları kullanılmıştır. Izzaralar ile bölünmüş hücrelerdeki en yüksek noktanın verisinin kullanılması ile 2.5D harita elde edilmesi sayesinde yükselti haritaları oluşturulmaktadır. Benzerlik metrikleri aracılığıyla ise bu haritadaki yükseklik bilgilerinin birbirine oturmasını sağlayacak konum ve yönelim farkı belirlenmektedir. Çalışmanın sonraki aşamalarında entropi teorisi kullanılması sebebiyle entropi temelli benzerlik metrikleri ile harita birleştirme yapılmıştır. Yedi farklı tipteki entropi metriği ile yapılan benzerlik karşılaştırması sonucunda “Jensen Divergence” entropi tanımının en az hata ile haritalar arasında dönme ve öteleme farkının belirlenmesini sağladığı, deneyler ile doğrulanmıştır. Ayrıca; haritanın dikey ekseninde katmanlara ayrılması ve bu katmanlar üzerinden yapılan yükseklik benzerlikleri hesaplaması ile optimum konum ve yönelim (veya dönme ve öteleme) farklarının belirlenmesinin; katmanlara ayırma metodunun kullanılmasına göre daha avantajlı olduğu da gösterilmiştir. Her bir otonom araç “Harita Birleştirme” süreci sonrasında bu harita Aktif-EZKH süreci için kullanılarak hem harita bilgileri daha hassas hale getirilir hem de robotun gitmesi gereken yeni konumu tespit edilmiş olur.

Yol planlaması, görevin etkin bir şekilde icrası için gerekli olan kritik adımlardan biridir. Enerji tüketim, elde edilen sonucun gerçekleşme süresi ve kalitesi uygulamanın ana kriterleridir. Bu nedenle, yol planlama algoritmaları etkin sistemler oluşturmak üzere kullanılmaktadır. Yol planlama algoritmaları farklı türde olabilir ama özellikle

hedef işaretleme ve bilgi maksimizasyonuna dayalı yöntemler diğer yol planlama yöntemlerine göre belirgin üstün özelliklere sahip olanlardır. Hedef odaklı yol planlama algoritmalarında, birimlerin belirli bir hedefe ulaşabilmesi adına oluşturduğu kontrol eylemleri bulunmaktadır. Bilgi maksimizasyonu yaklaşımı; ortam, nesnenin diğer nesnelere veya bir hedef hakkında daha fazla bilgi almak için bir doğrultu boyunca hareket etmesi olarak tarif edilebilir. Burada bağıl entropi teorisi, bilgi maksimizasyonu yaklaşımı olarak sunulmuştur. İlâveten, bağıl entropi, karşılıklı bilgi ile çevresel durum entropisiyle arasındaki farktır. Bağıl entropi kullanılarak, bilgi metrik olarak ifade edilebilmektedir. Çevresel durumlar ile gözlemler ile elde edilen durumlar arasındaki bağıl entropi üzerinden yaratılan amaç fonksiyonunun optimal çözümü sonucunda elde edilen hedef nokta, o bölgedeki bilginin belirlenen kriterlere göre istenilen seviyeye çekilmesini sağlamaktadır. Bu, EZKH ile etkileşimli çalışan yol planlaması temelli bir optimal kontrol yöntemidir. Bu yöntem çerçevesinde Bilgi Teorisinden faydalanılarak belirsizlik terimleri ile entropi terimleri arasında ilişki kuran bir Karşılıklı Bilgi terimi tanımlanır. Kulback-Liebler Mesafesi olarak da tanımlanan bu Karşılık Bilgi terimi maksimum değerine ulaştığında belirsizleri temsil eden entropi terimleri de minimize olurlar. Bu sebeple Karşılık Bilgi terimine dayalı bir amaç fonksiyonu oluşturularak bu fonksiyonu maksimize yapacak robot konum ve hareket vektörleri optimal kontrol yaklaşımı ile elde edilir. Bu elde edilen terimler heterojen robot takımında yer alan otonom robotlara uygulanarak onların hareketleri planlanmış olur. Amaç fonksiyonunu Lyapunov kararlı yapan bu noktalar ise bir hacimsel bölgenin merkezidir ve bu hacimsel bölgedeki bilgiyi maksimize etmek üzere belirlenmiştir. Bu noktaya ulaşmak için, robotlar belirlenen kurallar çerçevesinde hareket etmektedir. Bu kurallar ise İHA veya İKA'nın hedef noktaya hareketinin seçimi ve hedef noktaya ulaşım için engellerden kaçınmayı içermektedir. Bu yöntemin; özellikle farklı boyutlarda nokta bulutu ölçümü yapabilen hava ve kara araçlı robot takımındaki uygulamaları literatürde mevcut değildir. Bu teorik çalışmaları ön plana alan çalışmaların çıktılarının özellikle arama-kurtarma, keşif ve güvenlik gibi robot takımı uygulamaları için büyük önem taşıyacağı değerlendirilmektedir.

Önerilen yöntemde, ortamdan yapılan ölçümler ile araç hareketlerinde oluşabilecek belirsizliklerini etkilerini en aza indiren kara ve hava robotlarından oluşan heterojen yapı robot takımlarının keşif amaçlı yol planlama algoritmalarının geliştirilmesi ve performanslarının test edilmesi hedeflenmiştir. Aynı zamanda, bu görevleri icra edebilmek adına belirli harita birleştirmenin de gerçekleştirilmesi gerekmektedir. Öncelikle; harita birleştirilmesi yönteminin doğrulanması adına üniversite kampüsünde belirli bir bölgede kara aracı olarak Clearpath Husky A200, hava aracı olarak ise DJI Matrice 600Pro ve bu araçlar üzerinde bulunan Lidar sensörü kullanılmıştır. Sonuç olarak; teorik çalışmalarda verilen benzerlik metriklerinden en optimum olanı deneyler aracılığıyla belirlenmiştir. Sonrasında; bilgilendirici yol planlama yönteminin doğrulanması amacıyla Robot İşletim Sistemi ("ROS") ve Gazebo temelli, karmaşık ancak günlük yaşantıda karşılaşılabilen bir simülasyon ortamı kurulmuştur. Bu simülasyon ortamında altı farklı durum yaratılarak heterojen robot takımları için bilgilendirici yol planlamalı Aktif EZKH gösterilmiş ve parametre ayarlamaları ile uygulamaya göre değiştirilebilir bir yapı sağlanmıştır.



1. INTRODUCTION

Robots are commonly used in various applications, such as; target tracking, surveillance, exploration, etc. In these applications, complex problems are encountered to meet demands such as precise measurement, reconnaissance, eliminating uncertainties as much as possible, and so on. Heterogeneous robot teams (HeRTs) can be used to solve these problems with an optimal solution. Some of these applications can also be completed with a single robot or homogeneous robot teams (HoRT) of only ground vehicles or aerial vehicles. On the other hand, HeRTs can be employed to benefit each of the team member's different abilities. Unmanned ground vehicles (UGVs) can carry higher payloads such as larger batteries that can increase the operating time or more advanced Lidar devices. On the other hand, unmanned aerial vehicles (UAVs) can move faster and obtain environmental data from different angles, but they have limited endurance and capacity of payload. Naturally, extra challenges occur during heterogeneous team missions. One such challenge is acquiring a static global map that contains both information captured by UGV and UAV. There are two maps with different types of views, bird's-eye view and the ground view collected by the UAV and UGV, respectively. Both views have unique information, such as; the top of the objects or underneath the covered areas by a roof or tree. As a result, these two views offer more information about the environment and remove more uncertainties about the explored area by the robots. In the interest of acquiring a static global map, two maps need to be merged by using their common characteristics, such as the height of the sensed objects or other features in the environment.

Moreover, accomplishing the missions in more efficient ways in the manner of time, energy and accuracy are hard to achieve with a single robot. Hence, the use of multi-robot teams to get ahead that criteria are common nowadays. The employment for the missions may contain simultaneous actions as a robot's coordination and interaction. Fundamentally, navigation includes localization on the environment and mapping the environment. Simultaneous localization and mapping (SLAM) are specified for localizing the robot in the map and building the map of the robots' environment. The

robot builds a map by processing the sensor data and graph the environment. Localization is done by estimation within the evaluation of its position difference over the map and movement of the robot's itself. In the SLAM problem, better localization leads to better maps, also better mapping leads to better localization.

Additionally, for exploration missions, path planning is needed for efficient executions. Energy consumption, time and quality of the result are the main criteria of the application. Thus, path planning algorithms are used in order to create efficient systems. There are different types of path planning algorithms, but especially two main types distinguish such as goal designation and information maximization. In goal designated path planning algorithms, agents have a goal for the operation, and agents will have control actions to attain that goal. The information maximization approach can be described as; agent moves along a direction to get more information about the environment or other agents. Relative entropy theory provides a metric with the difference between mutual information and environmental state. The integration of SLAM and path planning in order to decrease uncertainty is named as Active SLAM. In particular, the robot moves towards other positions to get better localization and mapping with reducing uncertainty about the environment.

1.1 Purpose of Thesis

The main objective of this thesis is to design an active SLAM system with informative path planning for heterogeneous multi-robot teams. In active SLAM systems for HeRTs, there are three critical choices to achieve the goal;

- SLAM Method
- Map Merging Method
- Control Method

The selection of the SLAM method may vary in accordance with the system types. Some of the previous works in literature uses Gaussian distributions to evaluate posteriors. In order to provide simplicity and cover both systems in the same manner, EKF localization, LOAM, OctoMap and Height Map are employed. LOAM method with utilizing layers provides accurate localization over the map without the need for GPS. Height mapping ensures creating a common map by calculating the position and orientation difference with similarity metrics. EKF localization and OctoMap supplies

great localization and customizable map with exceptional visual outputs. These methods serve path planning for cooperative missions in this thesis.

In a cooperative framework, members of the team need to obtain information about the states of their own and the environment. To overcome this problem information maximization controlling method is used in the metric base. Relative entropy operates to measure information, which leads to maximizing information over the robot team. Robots' observations over the environment states are maximized by minimizing entropy with defining an objective function to control the system optimally. The proposed information maximization framework provides Lyapunov stable system for HeRTs. Also, control inputs ensure robots collision-free movement in the explored area.

1.2 Literature Review

In recent years, numerous studies have been conducted on robotic navigation and various types of applications such as search and rescue (Kumar et al., 2004), target tracking (Gorji et al., 2007), surveillance (Zhang et al., 2019), agriculture (Tokekar et al., 2016), and exploration of the environment (Makarenko et al.,s 2006).

Traditionally, in robotic navigation studies, SLAM as in; Thrun et al. (2005) is used to be aware of the agent's location and to understand the environment. Durrant-Whyte & Bailey (2006) presented different types of essential SLAM approaches such as EKF and Rao-Blacwellized Filter are compared. However, these SLAM algorithms require landmarks or GPS location to update the position. Also, similar to the approach presented by Kaess et al. (2012), these algorithms are used mostly in 2D mapping. Other than the essential SLAM methods, generalized iterative closest point (ICP) is provided by Segal et al. (2009) to map and localize without any additional sensors. However, the ICP method is not valid when the data rate of the Lidar sensor is slower than the motion of the robot. During the application given in this thesis, the aerial vehicle moves faster than the ground robot, and the rate of movement exceeds the data rates. In the research presented by Scherer et al. (2012), with the IMU sensor involved in the algorithm, distortions are eliminated when the aerial robot movements are relatively faster than the sensor data rates. Bosse & Zlot (2009) explain that a scan matching method similar to researches given by Yoshida & Tadokoro (2014) and

Bosse et al. (2012), which consists of spinning or moving a 2D laser measurement sensor to map the environment and localize the mobile platform.

Nevertheless, these methods are not suitable for real-time applications because of substantial post-processing requirements. To adopt this study for path planning in real-time, localization and mapping is needed. Kneip et al. (2011) achieved real-time visual odometry with a single camera and IMU with inherent difficulties due to the onboard camera view angle. Zhang & Singh (2014) have a solution to the real-time usage and employing heterogeneous agents problem in with the utilization of ICP based optimization method with a parallel algorithm that ensures the online update of the location and map. It suits both aerial and ground vehicles. The same approach in the application of this study is used to register the points and obtain the odometry data. With the LOAM method, robots can accomplish the missions without a need for GPS location data.

Although a single robot or robot teams can be utilized for robotic navigation, such as researches presented by Nurmaini & Tutuko (2017), Pham & Juang (2013), and Howard (2006), a HeRT can be implemented in the studies for a more detailed understanding of the environment. It is possible to employ HeRTs with a different point of view. An example of the implementation of HeRTs to the studies is given by Parker et al. (2004), in which the navigation problem is solved cooperatively. A robot team made up of identical mobile platforms carrying different types of onboard sensors can also be an HeRT mission. There is also a collaborative solution given by Hofmeister et al. (2011), where the parent robot operates child robots and gathering information from these robots with mapping the relative positions over the map. Heterogeneity can be defined as the ability to move on the ground, air, and sea. The study of Hood et al. (2017) presents a method that a UGV and UAV are used to explore the indoor environment and localize the robots by use of QR code, monocular camera, and Lidar. However, it is not aimed to get a merged map like in (Langerwisch et al., 2013). Common maps must be employed to use the map and odometry information in the same manner for both robots. In the interest of merging the maps, each robot can check common landmarks to reference maps by the relative positions to the landmarks as in (Ktiri & Inaba, 2012). Similarly, checking the occupancy state of the grids can be used for map merging as in (Husain et al., 2013). Yet, without the use of Lidar sensor, map merging is not suitable for long mission durations. Moreover, cooperative

monocular based SLAM for multi-UAV systems in a GPS denied environment is used without any heterogeneity of robots in (Trujillo et al., 2018). As stated by Fu et al. (2019), Lidars are more accurate than the camera. 3D Lidar sensors are utilized for the mission of this study to sense the environment with more accuracy and greater speed compared to the 2D and 1D types. Studies employing 3D Lidar sensors on both vehicles as in this study were not encountered during the literature survey except the study given in (Haddeler et al., 2020). However, demonstrations realized with only ICP type map merging method in that study.

The use of height values to merge maps and localize agents inside the map is handled in (Kleiner & Dornhege, 2007). Additionally, researches given by Nam et al. (2017), Forster et al. (2013), and Kaeslin et (2016) height maps are employed with different similarity metrics for collaboratively localizing the agents with only a small number of similarity metrics to find the position and orientation difference of two maps. This study becomes distinct from these techniques in the manner of map merging, where the aerial robot is utilized for assisting the ground vehicle pose estimation. Additionally, a common map approach with only three similarity metrics for a cooperative mission in a simulation environment is accomplished as in (Akay et al., 2018). Yet, the layering method is not employed to get a better map merging performance.

In this thesis, a unique solution that implements various entropy-based similarity metrics with the aim of constructing common maps of the environment with HeRTs is presented. In order to create common maps, Shannon entropy-based similarity metrics can be used, since it is the only one that holds the chain rule of conditional probability precisely. Seven distinct similarity metrics are compared, and the most effective one is chosen for getting a more comprehensive and valid common map. It is critical to acquire a proper common map for cooperative missions of aerial and ground agents because, without a valid common map, it is hard to plan trajectories for exploration of the environment. Moreover, different from all the studies in literature, the layering method is employed in order to compute the similarities of each local map obtained by a HeRT. This method also provides the accuracy of the merged common map since robots' sight of view prevents the same observations of the environment in features like a roofed top, forests, etc. This novel approach can also be used in GPS denied and

closed environments. The constructed common map allows creating an active SLAM system to explore the environment.

In studies provided by Carlone et al. (2010), Lázaro et al. (2013), Sileshi et al. (2013), Indelman et al. (2015), and Lourenco et al. (2015), the active SLAM with single robot approaches used in different ways, such as; particle-based SLAM posterior approximation, Markov Random Field approach, Independent Metropolis-Hastings Algorithm, dual-layer architecture and the Pontryagin minimum principle. Also, the active SLAM problem discussed in both ways, such as; a single robot and a robot team in (Julian et al., 2012). The execution of the mission or the necessity of the application may prompt to use multi-robot/agent teams similar to the studies as in; (Capitán et al., 2011) and (Indelman, 2015). Multi-robot teams are employed in order to get the results in less time, more accurate, and attain these performance outputs in more efficient ways. Similarly, along with the use of heterogeneous teams lead to efficient mission execution through the distinct abilities of the different kinds of unmanned vehicles as in (Sanfeliu & Andrade-Cetto, 2006) and (Capitán et al., 2011). To provide a more efficient outcome of the exploration mission from these studies, HeRT for active SLAM is also used in this study with a varied approach.

The planning method is one of the main criteria to create an active SLAM system. The common task of the path planning strategy is controlling the robot to change its orientation and localization to the desired direction. In literature, planning strategies under uncertainties can be separated in three ways, such as; look-ahead trajectory planning, informative path planning and simulation-based approach.

Informative path planning strategies or plans have constraints on time, fuel, energy, etc. Combinatorial optimization techniques used by Hollinger & Sukhatme (2013) are based on the increased available budget. The branch and bound approach is proposed by Binney & Sukhatme, 2012, while a Gaussian distributed model with mutual information theory is used in (Singh et al., 2009). The studies on informative path planning strategies were given before, assumes that the environment is known. However, in exploration missions, the environment cannot be known beforehand. The framework presented in this dissertation ensures that the only known parameter is the bounds of the explored area. In addition, the evaluation time increases with the scale of the problem. This study provides an optimum approach to adjust evaluation time in

accordance with the application type by employing Octree mapping techniques as in (Fairfield et al., 2007) and (Hornung et al., 2013).

Moreover, in simulation-based approaches, choosing the best strategy from the generated potential plans is one of the fundamental methods. The evaluation of the belief of the potential plans to quantify the quality is simulated in these approaches. Uncertainty metrics used in EKF-based planning analysis is provided by Lázaro et al. (2013). Similar to this, the EKF applied to the system as an inference engine in (Martinez-cantin et al., 2008), (Bryson & Sukkarieh, 2008) and (Martinez-Cantin et al., 2009). As a result, the EKF based localization method is also used for SLAM in simulation studies of this dissertation. Studies on simulation-based approaches assume maximum likelihood observations, by reason of future observations are given after the planning time, and the robots will receive the measurement assumption is defined.

Furthermore, look-ahead trajectory planning strategies evaluate the next given horizon control actions. Model predictive control strategy with EKF-SLAM is used by Huang et al. (2005) and Lourenco et al. (2015), the authors used the A-opt, D-opt and E-opt approaches to get over the optimization problem with the usage of Pontryagin minimum principle. Also, Sim & Roy (2005) used the A-opt in robot exploration applications, though all these studies are based on the discretization of the states. Handling the problem in a continuous manner is another challenging step in the path planning studies. Planning in the belief space (BS) refers, dealing with the problem in a continuous domain. Indelman et al. (2014) and Indelman et al. (2015) used BS for path planning for single and multi-robot teams. Regarding the robot number, the research provided by Indelman (2015) differs from the rest of the author's studies given in this literature review; however, the BS is still in use. Though, planning in BS strategy cannot provide the desired measurements in the manner of metrics of the objective. In order to illustrate the goal in the manner of metrics; some of the entropy approaches are presented in (Carlone et al., 2010), (Julian et al., 2012), (Atanasov et al., 2015) and (Meyer et al., 2015). In addition to that, Carlone et al. (2014) and Stachniss (2009) states that entropy-based approaches provide better outcomes than the frontier-based method given in (Yamauchi, 1998).

Relative entropy -also known as Kullback-Leibler divergence- approaches are widely applicable and have a great ability to solve complex combinatorial problems. Rao-Blackwellized particle filters are used for the estimation, but the study is only for single

robots as in (Carlone et al., 2010). On the other hand, Kontitsis et al. (2013) presented a framework with the multi-robot active SLAM system. Still, this study based on landmarks and EKF based SLAM method, and landmarks are not always located in the environment.

In conclusion, different from all the studies in literature, this dissertation provides an efficient and customizable framework to explore an unknown area with eliminating uncertainties. Also, the complexities of the multi-robot team missions are resolved by defined rules, similarity metrics, layering method and volumetric spaces. The theoretical background of constructing a common map for HeRTs is verified with experiments. Besides, the whole theory given in this dissertation is validated by the simulations in different cases.

1.3 Contributions

Main contributions of this study are listed below;

- Constructing common maps with the use of entropy-based similarity metrics and utilizing the layering method,
- Designing a controller for robots to maximize information and implementing cooperative estimation methods on the controller,
- Exploring the environment by employing HeRT with customizable mapping methods and adjustable information thresholds according to application.

1.4 Thesis Outline

The rest of the thesis is distributed as follows; the SLAM method in order to localize the robots in the environment and create the map for both systems. In subsection 2.1, the EKF localization algorithm is given to specify the system in a probabilistic manner. To ensure the HeRT's usability on GPS denied environments or without a sensor except for Lidar, the LOAM is given in subsection 2.2. OctoMap method implementation, details about the customizable parameters, and the effects of the parameters on mutual information are explained in subsection 2.3. Further, the height map technique is described, and the reason for employing the height map is given in subsection 2.3. After that, constructing common maps with utilizing the layering

method and heights of the objects are provided in section 3. The layering method for obtaining a more accurate and useful common map is stated in subsection 3.1. In the next subsection (3.4), the seven distinct similarity metric which are based on entropy theory is given. Section 4 provides the main contribution of this dissertation as designing an optimal controller (subsection 4.2) in accordance with the relative entropy theory (subsection 4.1). The results of demonstrated real-time experimental and simulation studies for the realization of the theoretical background given in this dissertation are presented in section 5. In subsection 5.1, the experimental studies of constructing a common height map are carried. After that, in subsection 5.2, the demonstration of the simulations and results are provided. Lastly, to discuss the outcomes and summarize the thesis, the conclusion is given in section 6.





2. SLAM METHOD

In this section, the method for localization of the mobile robots and mapping of the environment is explained. First, the theoretical background of the EKF Localization for UGV and UAV with proprioceptive sensors, such as; IMU and environmental sensor as a GPS carried for both robot is given briefly to determine the location of the robots in the environment. The second part is allocated for the LOAM method, which is also a localization method for mobile robots without any environmental sensors. This method is employed for the real-time experimental validations to prevent the inaccurate georeferencing of the point clouds due to the sensor noises. After that, to visualize the environment in 3D with less memory usage, Octree Mapping or OctoMap method is presented. In the last part of this section, the height mapping of the environment to understand the surround in the same manner for both robots and construct common maps with similarity metric, which is given in section 3.

2.1 EKF Localization

EKF is a Gaussian filter used for nonlinear systems such as mobile robots in order to estimate their states, such as; pose and velocity in continuous operations. Localization of the mobile robots carrying onboard GPS and IMU is done by employing extended Kalman Filter in the simulation case studies because of its simplicity and effective computational performance.

The well-known algorithm of the EKF has two parts; prediction and correction. Prediction algorithm of the robot state with discrete time intervals, k , is given as;

$$\bar{X}_k = f(X_{k-1}) \quad (2.1)$$

$$\bar{P}_k = F_k P_{k-1} F_k^T + R_k \quad (2.2)$$

Where \bar{X}_k is the predicted state of the robot, f is the state function and x_{k-1} is the previous state of the robot. x_k is the robot's state comprises its 3D pose, orientation and velocities. State vectors of UGV and UAV include 3D position, velocity, and orientation can be written as;

$$X_k = [x_k \ y_k \ z_k \ \dot{x}_k \ \dot{y}_k \ \dot{z}_k \ \varphi \ \theta \ \psi]^T \quad (2.3)$$

First three elements of the X_k vector refers to the position of the robots. The fourth to sixth elements stands for the velocity, and the GPS sensor measures these six states. The last three of the state vector X_k , are the orientation in Euler Angles, which are measured by the IMU sensor. Without the GPS and IMU sensor noises, it is possible to obtain states of the robots in a deterministic way. However, these sensors have continuous drifts and additive noise in the real world. Despite the fact that EKF localization is used for simulation cases in this thesis, it is better to add disturbances (noises) in the calculation to converge the real ones.

\bar{P}_k denotes predicted covariance, F_k is Jacobian of f as a nine by nine matrix. P_{k-1} is the known covariance at discrete time $k - 1$. The covariance of Gaussian random vector (process noise) is denoted as R_k .

The correction step is given below;

$$K_k = \bar{P}_k H_k^T (H_k \bar{P}_k H_k^T + Q_k)^{-1} \quad (2.4)$$

$$x_k = \bar{x}_k + K_k (m_k - H(\bar{x}_k)) \quad (2.5)$$

$$P_k = (I - K_k H_k) \bar{P}_k \quad (2.6)$$

Here, K_k is the Kalman gain, and H_k is the Jacobian matrix of the observation function h . m_k denotes measurements at time interval k and it contains states collected by GPS and IMU sensors, Q_k is for observation noise, and I is the nine by nine identity matrix.

Corrected robots' states and covariance are obtained with equation (2.5) and (2.6) at time interval k . After that, for the next time steps, states of the robots are calculated with the continuous loop starting from equation (2.1) to (2.6).

As a result of the EKF localization, it possible to georeference the Lidar point clouds with the rotation and translation vectors defined by the use of laser measurement sensor position and orientation difference from the onboard IMU and GPS sensors.

2.2 Lidar Odometry and Mapping

In order to obtain a globally referenced map of the surroundings for Lidar mounted vehicle practices, all the sensed points must be designated. If there will not be any designation of the points, laser measurements may attach to the same points or disruption may occur. As a result, point clouds seem unrecognizable, and the area

cannot be classified for the aimed results. Aerial and ground robots must georeference the points to provide a globally referenced local maps. After that, grid maps handled with georeferenced point cloud to enable better representations of the area by employing heights of the objects on the cells.

Lidar Odometry method given by Zhang & Singh (2014) is used to reference the points measured by Lidar sensors. This approach provides receiving georeferenced map and location of the agents without any other sensors, such as; IMU, GPS, and encoder.

In this study, VLP-16 Lidar is employed to sense the objects and the surroundings. This sensor has 16 optical scanners rotating around its vertical axis at a maximum of 3000rpm. Besides, these scanners are located vertically in order to sense the environment horizontally $\pm 15^\circ$ from the center point. $n, n \in \mathbb{Z}^+$ refers optical channel and the map obtained by each channel can be shown by S_n . L is the coordinate system of Lidar and coordinates of the point in $L_n, i, i \in S_n$ the point cloud is shown by $C_{(n,i)}^L$. $\{D\}$ is the referenced coordinate system and $C_{(n,i)}^D$ is the point inside that coordinate system. Also, G is the set of points obtained during a scan by n optical channel. The surface roughness metric κ must be derived to implement Lidar Odometry in the system of this thesis using the formulation below;

$$\kappa = \frac{1}{|G| \times \|C_{(n,i)}^L\|} \|\sum_{j \in G, j \neq i} (C_{(n,i)}^L - C_{(n,j)}^L)\| \quad (2.7)$$

This metric enables us to sort measured points, whether these are edge or surface (Figure 2.1). If κ as its maximum values or above the threshold measured points can be sorted as edge Γ_n . Otherwise, where the κ value is below the threshold, these points are surface points, Δ_n . That information allows finding position difference by using Euclidian distances.

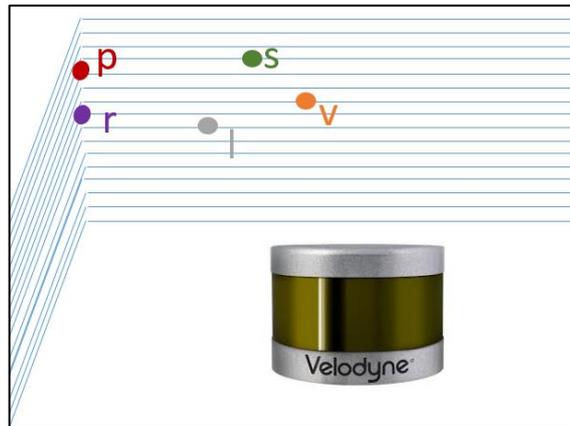


Figure 2.1 : Edge (p and r) and surface (s,v and l) points.

The distances between different time interval scans S_n and S_{n+1} is;

$$d_\Gamma = \frac{|(c_{(n+1,i)}^L - c_{(n,j)}^L) \times (c_{(n+1,i)}^L - c_{(n,l)}^L)|}{c_{(n,j)}^L - c_{(n,l)}^L} \quad (2.8)$$

Here, i is the nearest point to the edge point i , $i \in \Gamma_k$, l is the sensed point from the neighbour optical channel and d_Γ is the distance from edge points to the line which is created within (j, l) , $j, l \in S_k$ points.

Moreover, point to surface distance will be;

$$d_\Delta = \frac{|(c_{(n+1,i)}^L - c_{(n,j)}^L) \times (c_{(n,j)}^L - c_{(n,l)}^L)| \times (c_{(n,j)}^L - c_{(n,m)}^L)}{|(c_{(n,j)}^L - c_{(n,l)}^L) \times (c_{(n,j)}^L - c_{(n,m)}^L)|} \quad (2.9)$$

where i is the nearest point of surface point i , $i \in \Delta_n$, l is the second nearest point which is sensed from the same optical channel and $j, l \in S_n$. The sensed point m from the neighbor optical channel $m \in S_{n+1}$ and d_Δ is the distance from surface points to the surface which is created within (j, l, m) , $j, l, m \in S_n$ points.

Let t_σ is the duration of $0 \rightarrow 2\pi$ scan of the 3D Lidar and t is the moment at the investigation of position and localization difference $T_\sigma^L = [p_x, p_y, p_z, \theta_x, \theta_y, \theta_z]^T$. The position and location difference between $[t_i, t_\sigma]$ is;

$$T_{\sigma,i}^L = \frac{t_i - t_\sigma}{t - t_\sigma} T_\sigma^L \quad (2.10)$$

3D Lidar which holds 16 optical channel, measures the environment in δ_z time steps.

So the rotation and translation terms can be written as;

$$C_{(n,j)_t}^L = RC_{(n,j)_{t-\delta_z}}^L + T_{\sigma,i}^L \quad (2.11)$$

where the rotation matrix is indicated as R . In this case features of edge and surface geometric relationship can be written as functions given below;

$$f_\Gamma(C_{(n,j)_t}^L, T_{\sigma,i}^L) = d_\Gamma, \quad i \in \Gamma_n \quad (2.12)$$

$$f_\Delta(C_{(n,j)_t}^L, T_{\sigma,i}^L) = d_\Delta, \quad i \in \Delta_n \quad (2.13)$$

Finally, to obtain minimized function for each feature point nonlinear optimization method is handled;

$$\min(T_{\sigma,i}^L - (J^T J + \lambda \text{diag}(J^T J)) J^T d) \quad (2.14)$$

where $f(T_{\sigma,i}^L) = d$ is the nonlinear distance, $J = \delta f / \delta T_{\sigma,i}^L$ and λ is a factor provided by the Levenberg-Marquardt method. $T_{\sigma,i}^L$ is found by minimizing the distance to zero

and the georeferenced point cloud with registering the points $C_{(n,j)_t}^L$ at each time step by the following operation;

$$M_{LOAM_t} = \sum_{t=0}^t RC_{(n,j)_{t-\delta_z}}^L + T_{\sigma,i}^L \quad (2.15)$$

is obtained. After that, OctoMap can be handled within the usage of $C_{(n,j)_t}^L = m_t$.

2.3 OctoMap

OctoMap is a great visualization tool for 3D environments with computational efficiency and flexible structure for resolution and explored environment (Hornung et al., 2013). In simple terms, OctoMap is a mapping technique similar to 2D occupancy grid mapping. It is dealing with 3D volumetric occupancies instead of occupied 2D planes as it can be seen in Figure 2.2

OctoMap technique employs the octrees, which are hierarchical data structures in accordance with the decomposition of space (Mao et al., 1987). The space is hierarchically divided into eight subvolumes, and the whole volume refers to the first element of the octree, which is called the root (Wilhelms & Van Gelder, 1992). The occupancy probability of the detected points that will be transformed as a subvolume called voxels can be written with the log-odds presentation as which is similar to (Li & Ruichek, 2014);

$$l(V_i|m_{1:t}) = l(V_i|m_{1:t-1}) + l(V_i|m_t) \quad (2.16)$$

Where;

$$l(V_i|m_t) = \log \left(\frac{P(V_i|m_t)}{1 - P(V_i|m_t)} \right) \quad (2.17)$$

Here, V_i is the voxel investigated whether occupied or free, m_t is the collected measurements, which are point clouds. $l(V|m_{1:t})$ is the joint probability of voxels' occupancy state with the log-odds presentation, and specifies the update rule for mapping. However, it must be considered to insert a limitation to updates in equation (2.16) to decrease the computational effort for updating the occupancy state of the voxel.

Limiting the number of updates can be done by giving boundary values of $l(V_i|m_t)$ which can be denoted as the lower bound and upper bound of OctoMap, recursively;

l_{LB_o} and l_{UB_o} . As a result of that, voxels are stated as permanently occupied when both of the boundary values are reached. So, the equation (2.16) evolves to;

$$l(V_i|m_{1:t}) = \max(\min(l(V_i|m_{1:t-1}) + l(V_i|m_t), l_{UB_o}), l_{LB_o}) \quad (2.18)$$

l_{LB_o} and l_{UB_o} values are determined by the information threshold on the explored area, which is going to be explained in section 4.

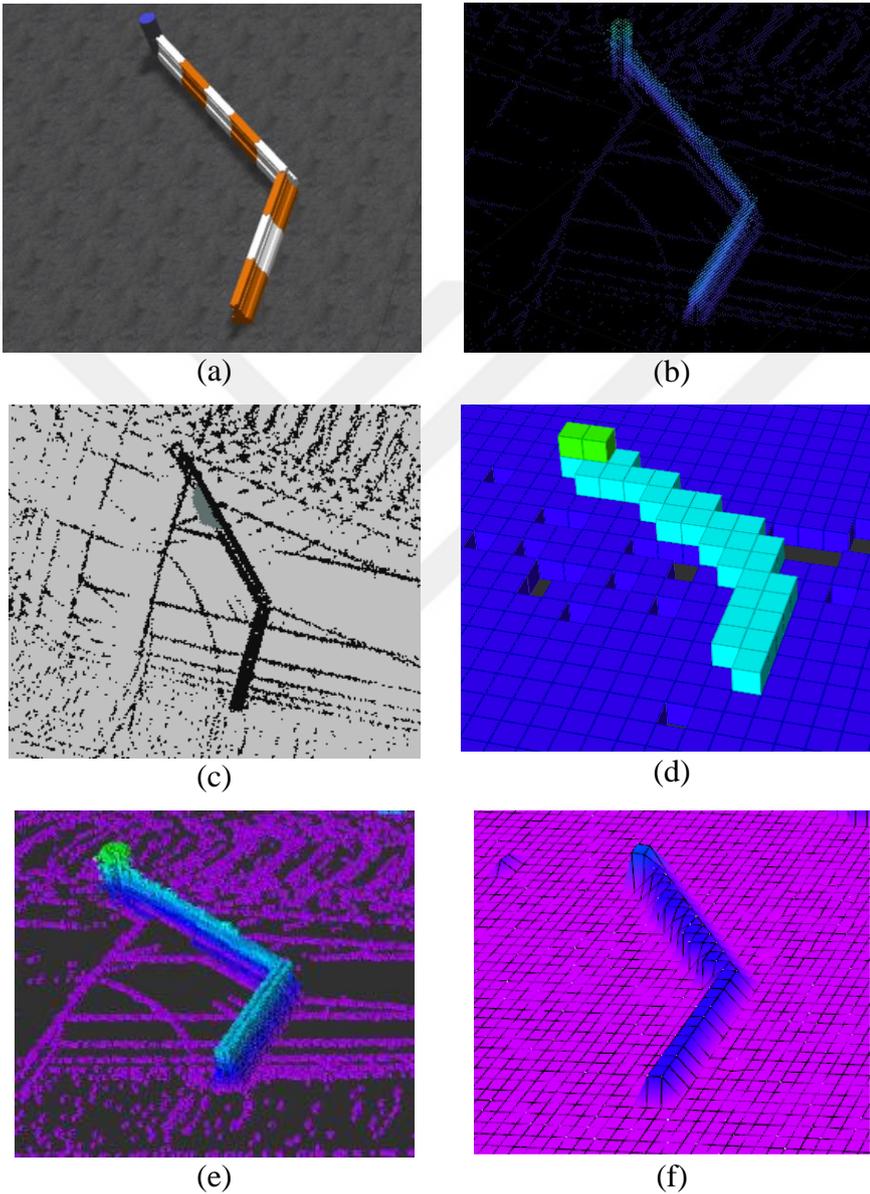


Figure 2.2 : Different types of mapping results. (a) a simulation scene element barrier from Gazebo. (b) Point cloud representation. (c) 2D projected grid map. (d) OctoMap with resolution = 1. (e) OctoMap with resolution = 0.05. (f) Elevation map.

The probability of the occupied voxels in the map differs between 0 to 1. If the limits determined in accordance with $P(V_i|m_t) = [0: 1]$, there will not be any permanently

occupied cells, but without any loss of information. The limit values near $P(V_i|m_t) = 0.5$ such as $l_{LB_o} = -0.2$ and $l_{UB_o} = 0.2$, cause loss of information about the environmental map, M_e , though it leads less memory usage by decreasing measurement update numbers for determining permanently occupied cells. Changing limit values is one of the flexible properties of the OctoMap technique. The second one is obtaining the map with different resolution values that can provide coarse or precise path planning control inputs in the robot's local map to avoid the obstacles. Assigning averages of each inner voxels' log-odds occupancy value to the root node will offer less use of memory than the maximum values of each inner voxels. On the other hand, choosing the maximum values of each inner voxels as the root log-odds occupancy value assists the precise path planning without any collisions. In conclusion, the assignment of inner voxels' occupancy probabilities to the root voxel and the limit boundary values may vary depending on the application, which will be given in section 5.

2.4 Height Mapping

Map attained from an aerial robot is like a bird's-eye view and has difficulties in capturing vertical surfaces. On the other hand, the ground robot has challenges in getting the view of the area behind the taller obstacles and the depth dimension of these obstacles.

In Figure 2.3, the sight of view of HeRT team members can be seen. In this scene, the shortcomings of each agent's sensors in the complex environment are exemplified. As it can be seen from the left and right side of Figure 2.3.a, the aerial robot could not detect objects and surface details under horizontal wall such as litter bin no.1 and table. On the other hand, litter bin no.2 is perceived by the aerial robot in contrast to the ground robot. Also, the ground robot is able to sense litter bin no.1, table, and the inner surface of the vertical wall. Therefore, maps obtained from vehicles do not look alike. The height mapping method is employed to resolve this issue.

Height maps can be defined as 2.5D grid map representation of the area. 2.5D grid map representation, as in Figure 2.4, provides better calculation time and less storage need compared to using 3D maps. Also, 2.5D grid maps enable comparing grids received by vehicles through the use of height information. Height maps can be easily

implemented to the study since local point cloud maps are obtained beforehand. In its basic form, the map is divided into 2D grids, and the point with the maximum value in the vertical axis is assigned as the height of each grid.

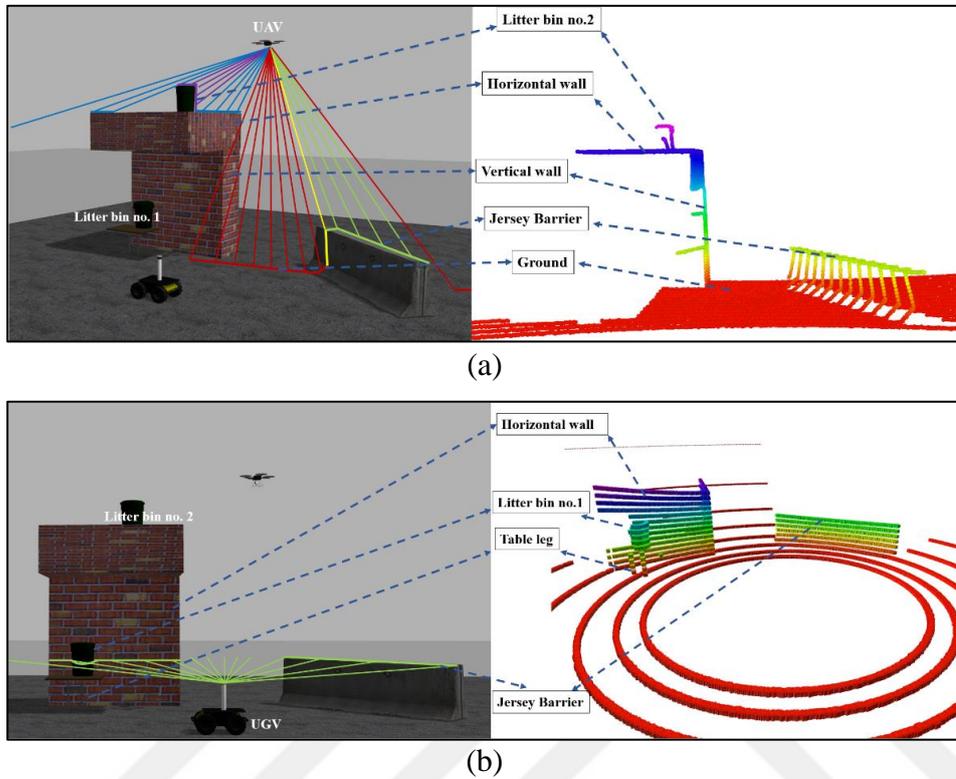


Figure 2.3 : Lines with different colours show the representative laser beams for one channel. Distances to objects are illustrated by colours of the beams on the left side of the figures. Colour is changing with respect to distance. For better understanding, robots are omitted from point clouds on the right side of the figures. (a) Aerial robot’s representative sight of view and the point cloud respectively left and right side. (b) Ground robot’s representative sight of view and the point cloud respectively left and right side.

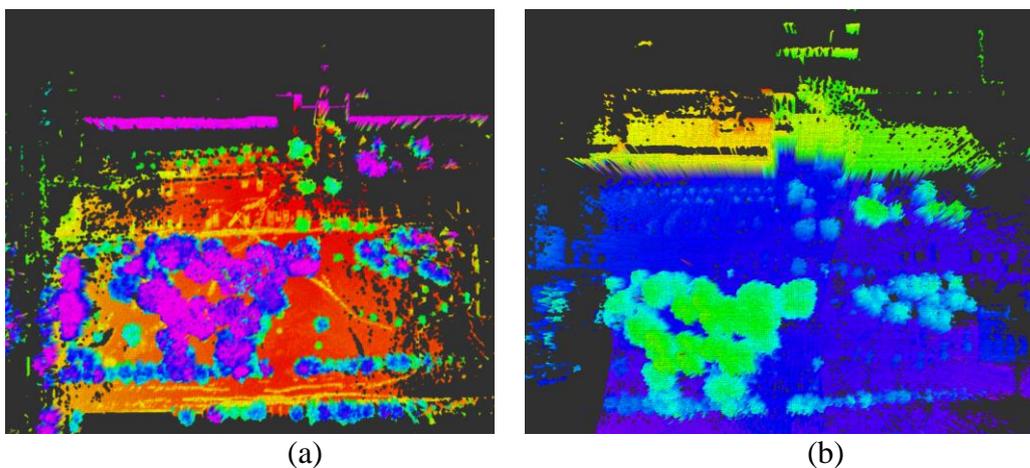


Figure 2.4 : Height map 2.5D X-Y plane projection. (a) Ground robot’s local height map. (b) Aerial robot’s local height map.

3D illustration with the isometric view of the height maps are given in Figure 2.5

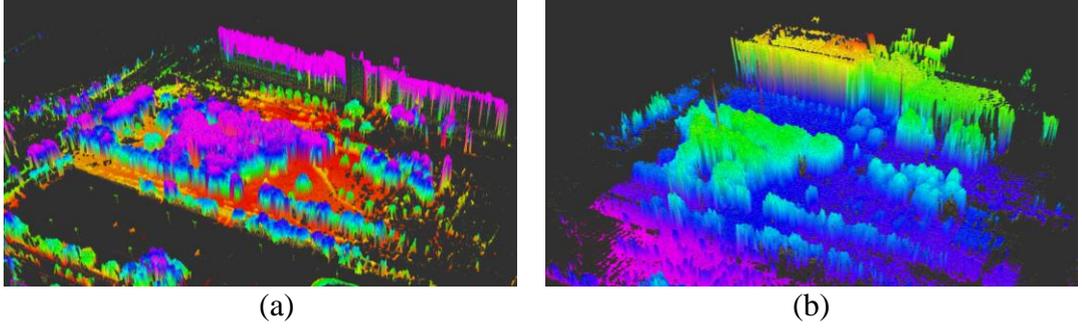


Figure 2.5 : Height map 3D illustration. (a) Ground robot's local height map. (b) Aerial robot's local height map.

The height map \mathbf{H} can be expressed as in (Yang & Wang, 2011) ;

$$\mathbf{H}_{0:t} = \{(O_{0:t}, \mu_{0:t}, \sigma_{0:t}^2)_i, i = 1, \dots, n\} \quad (2.19)$$

Here, $O_{0:t}$ refers to the occupancy probability of each grid g_i from beginning to time t . $O_{0:t}$ is calculated by log-odds presentation similar to equation (2.16), except this probability value investigates occupancy of 2D planes. $\mu_{0:t}$ is the height estimation and $\sigma_{0:t}^2$ is the variance of the height.

In this dissertation, height maps employed after obtaining georeferenced point clouds by EKF localization, LOAM and OctoMap, hence the sensor measurement noises are omitted in $\mathbf{H}_{0:t}$ calculation. As a result of that, $\mu_{0:t} = z_t$ and $\sigma_{0:t}^2 = 0$ can be written.

Finally, the occupancy probability of the cell with log-odds presentation can be defined as;

$$l(g_i|m_{1:t}) = \max(\min(l(g_i|m_{1:t-1}) + l(g_i|m_t), l_{UB_h}), l_{LB_h}) \quad (2.20)$$

With;

$$l(g_i|m_{1:t}) = \log \left(\frac{P(g_i|m_t)}{1 - P(g_i|m_t)} \right) \quad (2.21)$$

l_{LB_h} and l_{UB_h} is the lower and upper bound of the log-odd value to limit the update rate of the occupancy state of the grid. Thus, if the occupancy probability value is higher than the threshold for the grid i , the height value of the grid is assigned as the maximum height value of the georeferenced point cloud on that grid as in equation (2.22).

$$z_t = \begin{cases} z_i, & \text{if } P(g_i|m_t) > l_{th} \\ 0, & \text{if } P(g_i|m_t) \leq l_{th} \end{cases} \quad (2.22)$$

Where z_i is the value of the point in the coordinate frame z in each cell, and the l_{th} is the occupancy probability threshold value.

In conclusion, the localization method and various types of mapping techniques are given. The EKF localization is a simple and easily implementable method that is used in simulation studies with GPS and IMU sensors are in the loop. However, when GPS signal is denied or the sensor is not used onboard LOAM method makes the system utilizable for the exploration mission. OctoMap method provides an adjustable framework in accordance with the information seeking controller. Also, it ensures to the variance of the memory usage property with resolution adjustment. Further, the height mapping technique offers a map merging infrastructure. With the obtained local height map of each robot, it can be separated into the layers to understand the environment for robot movement and to get a precise common map using the similarity metrics, which is explained in the next section.

3. CONSTRUCTING COMMON MAP

In cooperated or collaborated applications for HeRTs, there will be different local maps which are obtained separately. The local maps acquired by the use of EKF or LOAM may not have the same position and orientation for each robot in a global coordinate system. Despite the fact that EKF employs GPS and IMU, drifts may occur in georeferencing the point clouds to the global origin because both sensors have bias values, and they differ for each robot. For LOAM, there is not any environmental sensor employed for mapping. As a result, the local maps' coordinate systems of each robot have position and orientation differences.

Further, the primary purpose of this study is to maximize information about the environment with HeRT. Each team member can maximize the information on its own without checking the area is scanned by another member or not. However, this procedure is not effective for exploration missions, since the information about the investigated area may also be obtained.

Constructing a common or joint map for team members will overcome these problems. Similarity metrics can be employed in order to merge local maps of the robots. The working principle of the common map construction method can be seen in Figure 3.1

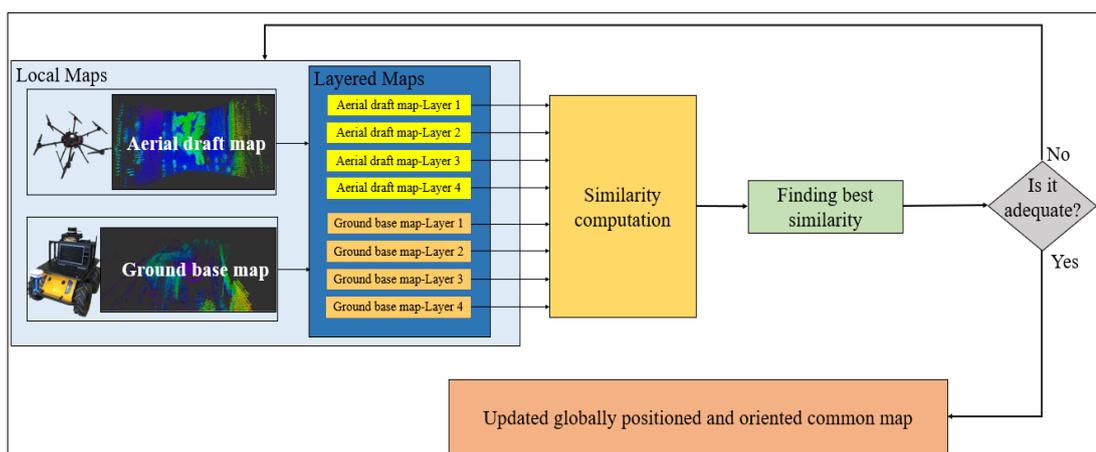


Figure 3.1 : Working principle of common map construction.

The height values in local maps' layers attained by UAV and UGV are used in the similarity computation process to find the best similarity value in terms of position and orientation between aerial draft and ground base maps. If the similarity value is

maximized (or distance/entropy value is minimized) enough for each layer, the common map can be constructed and updated by time intervals as in (Akay & Temeltaş, 2020). Further, the layering method and various similarity metrics and their usage in this study are explained below.

3.1 Layering Method

In this dissertation, the main purpose of obtaining a common map is to benefit all the agents in the mission. To classify or define a path for the ground robot, it is not necessary to obtain any information about the obstacles located above its height. In contrast, aerial robots demand information about features at all altitudes in the environment since these robots can fly at different altitudes within their limits. Moreover, heights of the same features in the environment may be measured differently due to the robots' sight of view. In order to overcome this problem, the layering method is employed in this study.

Additionally, the layering method will be a useful tool for information seeking path planning. In this study, collected information about the environment is investigated in volumetric spaces. Detailed explanations will be given in section 4.

For the layering method, layers in the z-direction in relation to the vehicle's features are formed. Layer 1 has information about the area where $z < h_1$, layer 2 has information about the area where $h_2 < z < h_3$, layer 3 has information about the area where $h_3 < z < h_4$ and layer 4,5,...,n has information about the area where $h_4 < z < h_4 + (n - 4) h_{uav}$. Heights of the layers can be seen in Figure 3.2.



Figure 3.2 : A representative figure of the heights of the layers from the experimental study used in this dissertation.

3.1.1 Layered Map of UGV

The first layer is between the base of the environment and the second layer. This layer has the height information of the obstacles that UGV can climb over. In this study, the HUSKY A200 vehicle will be used as UGV, which can climb obstacles less than 150mm height from the base. The third layer's height is aligned with UGV's maximum height. The height of HUSKY A200 with the Velodyne VLP16 mounted on a beam was measured as 475mm from the ground. Layers of the local maps perceived by the ground robot can be seen in Figure 3.3.

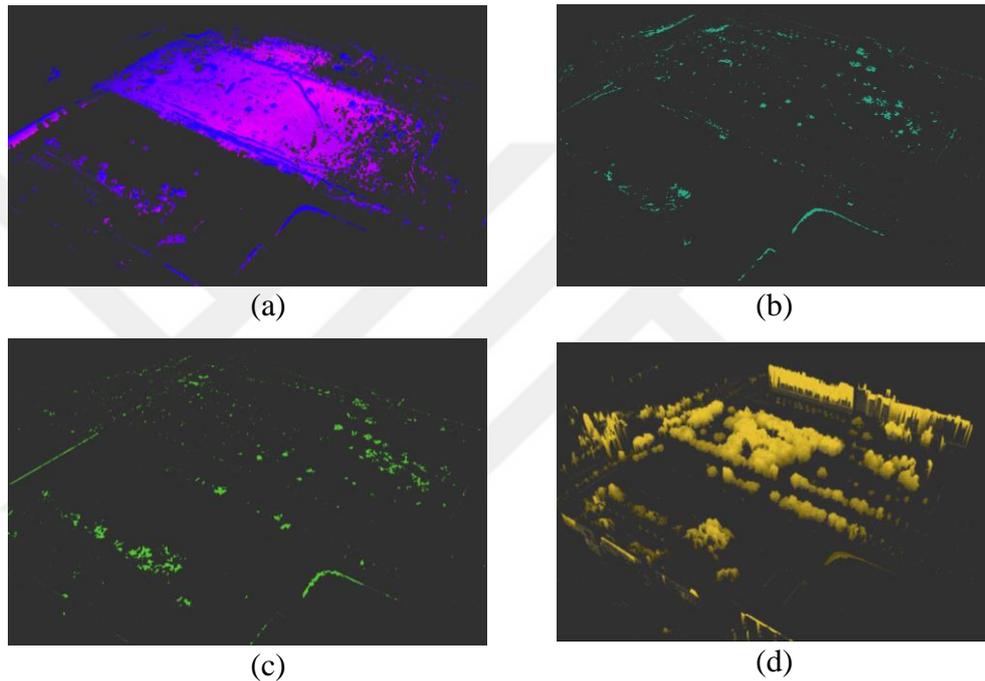


Figure 3.3 : UGV layered local height maps. (a) 1st layer. (b) 2nd layer. (c) 3rd layer. (d) 4th layer.

3.1.2 Layered Map of UAV

The third layer and the layers above this layer has the height more than aerial vehicles' height, $h_3 = 450mm$. This height value is chosen for the layer, since UAV is interested in the layers that it can pass through. For this dissertation, $h_4 = 39075mm$ is chosen to decrease the complexity of the calculations.

Layers of the local maps which are perceived by the aerial robot can be seen in Figure 3.4. Note that, different colours of the layers are selected to identify them well in both Figure 3.3 and Figure 3.4. For the first layer, blue refers to the heights of the objects, and magenta refers to the base of the investigated area. Aqua colour is chosen for the

second layer, green colour refers to the third layer, and for the fourth layer, yellow is selected.

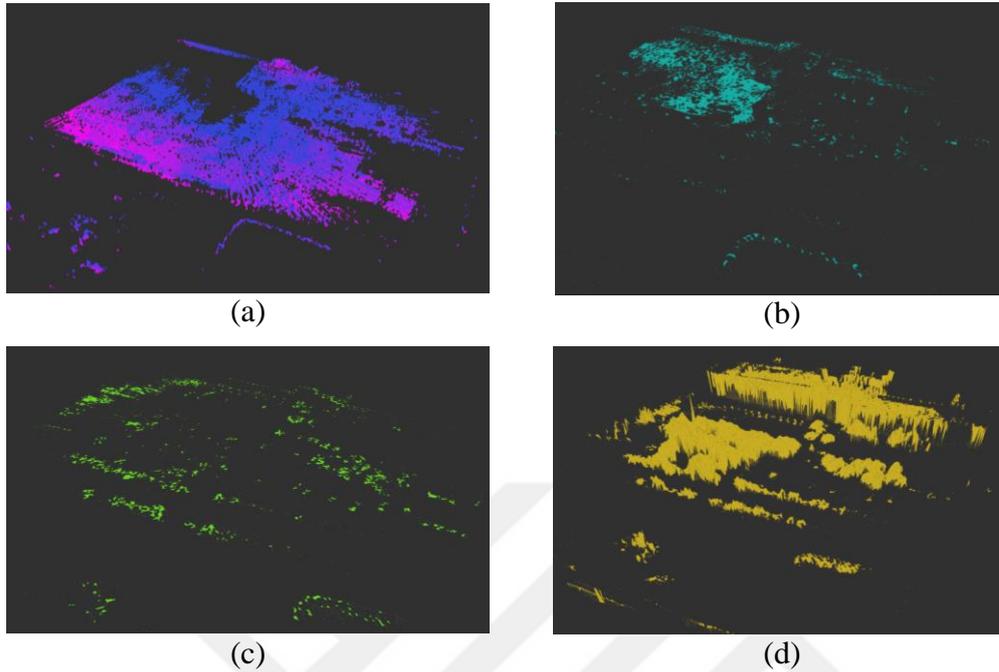


Figure 3.4 : UAV layered local height maps. (a) 1st layer. (b) 2nd layer. (c) 3rd layer. (d) 4th layer.

Similarity metrics are employed for each layer to get common maps for each layer.

3.2 Similarity Metrics

Similarity measures are encountered in various types of disciplines, such as mathematics, biology, economy, physics, information theory, ecology, etc. For any applications, different types of similarity measures can be utilized. The distance between each other defines the similarity of objects. Similarity measures can be mainly classified as vectorial and probabilistic. Vectorial measures rely on overlapping two vectors in terms of distance, while probabilistic measures are calculated by empirical estimations, such as entropy. Kullback-Liebler Divergence (KLD), Jeffrey Divergence (JD), K Divergence (KD), Topsoe Divergence (TD), Jensen-Shannon Divergence (JeSD) and Jensen Divergence (JeD) used as the similarity metrics of this dissertation. These metrics are derived from the Shannon entropy (SE) definition (Cha, 2007).

In this study the entropy definitions mentioned before are used as follows; the difference of position and orientation, respectively translation and rotation between local maps are denoted as $u = \{u_1, \dots, u_n\}$ or T, R and $u_i = (x_i, y_i, \theta_i)$. Where u_i

contains distances in 2D x_i, y_i (translation, T) and angular difference as θ_i (rotation R). The best position and orientation \tilde{u} value can be found by minimizing entropies or distance values obtained by different entropy approaches stated below.

SE definition is as follows;

$$H_{bd}(u) = - \sum_{i_b=1}^{n_b} \sum_{i_d=1}^{n_d} p_z(z_b, z_d|u) \log_2 p_z(z_b, z_d|u) \quad (3.1)$$

And the best position and orientation with regard to SE definition is;

$$\tilde{u} = \arg \min_u H_{bd}(u) \quad (3.2)$$

KLD definition can be given as;

$$d_{bd}(u)_{KLD} = \sum_{i=1}^d p_z(z_b|u) \ln \frac{p_z(z_b|u)}{p_z(z_d|u)} \quad (3.3)$$

And the best position and orientation concerning KLD definition is;

$$\tilde{u} = \arg \min_u d_{bd}(u)_{KL} \quad (3.4)$$

The third one, JD is as follows;

$$d_{bd}(u)_{JD} = \sum_{i=1}^d (p_z(z_b|u) - p_z(z_d|u)) \ln \frac{p_z(z_b|u)}{p_z(z_d|u)} \quad (3.5)$$

And the best position and orientation with regard to JD definition is;

$$\tilde{u} = \arg \min_u d_{bd}(u)_{JD} \quad (3.6)$$

The fourth entropy definition, KD is given as;

$$d_{bd}(u)_{KD} = \sum_{i=1}^d p_z(z_b|u) \ln \frac{2p_z(z_b|u)}{p_z(z_b|u) + p_z(z_d|u)} \quad (3.7)$$

And the best position and orientation concerning KD definition is;

$$\tilde{u} = \arg \min_u d_{bd}(u)_K \quad (3.8)$$

After that the TD definition is as follows;

$$\begin{aligned} d_{bd}(u)_{TD} = & \sum_{i=1}^d p_z(z_b|u) \ln \frac{2p_z(z_b|u)}{p_z(z_b|u) + p_z(z_d|u)} \\ & + p_z(z_d|u) \ln \frac{2p_z(z_d|u)}{p_z(z_b|u) + p_z(z_d|u)} \end{aligned} \quad (3.9)$$

And the best position and orientation with regard to TD definition is;

$$\tilde{u} = \arg \min_u d_{bd}(u)_T \quad (3.10)$$

The JeSD formulation, which is similar to JeSD, can be given as;

$$d_{bd}(u)_{JeSD} = \frac{1}{2} \left[\sum_{i=1}^d p_z(z_b|u) \ln \frac{2p_z(z_b|u)}{p_z(z_b|u) + p_z(z_d|u)} + p_z(z_d|u) \ln \frac{2p_z(z_d|u)}{p_z(z_b|u) + p_z(z_d|u)} \right] \quad (3.11)$$

And the best position and orientation concerning JeSD definition is;

$$\tilde{u} = \arg \min_u d_{bd}(u)_{JeSD} \quad (3.12)$$

Lastly, the JeD formulation is given as follows;

$$d_{bd}(u)_{JeD} = \frac{1}{2} \left[\sum_{i=1}^d p_z(z_b|u) \ln p_z(z_b|u) + p_z(z_d|u) \ln p_z(z_d|u) - (p_z(z_b|u) + p_z(z_d|u)) \ln \frac{p_z(z_b|u) + p_z(z_d|u)}{2} \right] \quad (3.13)$$

And the best position and orientation with regard to JeD definition is;

$$\tilde{u} = \arg \min_u d_{bd}(u)_{JeD} \quad (3.14)$$

Where, z_b and z_d denote the height values of the objects in local maps captured from UGV and UAV, respectively. $p_z(z_b|u)$ and $p_z(z_d|u)$ refers to the joint probability of the z_b and z_d height values with the rotation and translation implemented to the obtained height map. u is applied to the aerial and referenced height map of the environment $A_m(z)$ and $G_m(z)$ respectively with;

$$G_m(z) = RA_m(z) + T \quad (3.15)$$

Here,

$$u = \begin{bmatrix} R_{2 \times 2} & T_{2 \times 1} \\ 0 & 1 \end{bmatrix} \quad (3.16)$$

With,

$$R_{2 \times 2} = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix} \quad (3.17)$$

$$T_{2 \times 1} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (3.18)$$

Distance values are obtained by summation of the joint probabilities with given formulations and the best orientation, \tilde{u} is calculated with minimizing the distance or entropy functions given above. Affine 3D transformation matrix u is iteratively applied to equation (3.15) for minimizing the distance and entropy functions.

As a result, the position and orientation difference between aerial draft map layers and ground base map layers are found. The common map, M_{w_c} is obtained by summation of the transformed aerial local map A_m and ground map G_m ; $M_{w_c} = G_m + A_m$. This merged map provides a better understanding of the environment and a background to calculate uncertainties/entropies between observed states of each robot in a common ground. Without the common map, each robot may observe the same points in the environment. Hence the controller will not be optimal enough for exploration.





4. INFORMATIVE PATH PLANNING

The aim of the exploration missions is to collect information about the designated area. The amount of information collected by robots must be known to define the success criteria in autonomous missions. Relative entropy theory is utilized as an information metric in this dissertation. Information and relative entropy have a strong relation, in which information is maximized by minimizing the relative entropy between observations and environmental states. The difference between the collected information by robots and all the information that can be collected provides control inputs for robots. Within that control input and collision-free path planner, robots will obtain measurements at the next position and determine the updated relative entropy between environmental states and observed states, as shown in Figure 4.1. After that, the continuous operation will proceed until the observed states converge to the environmental states.

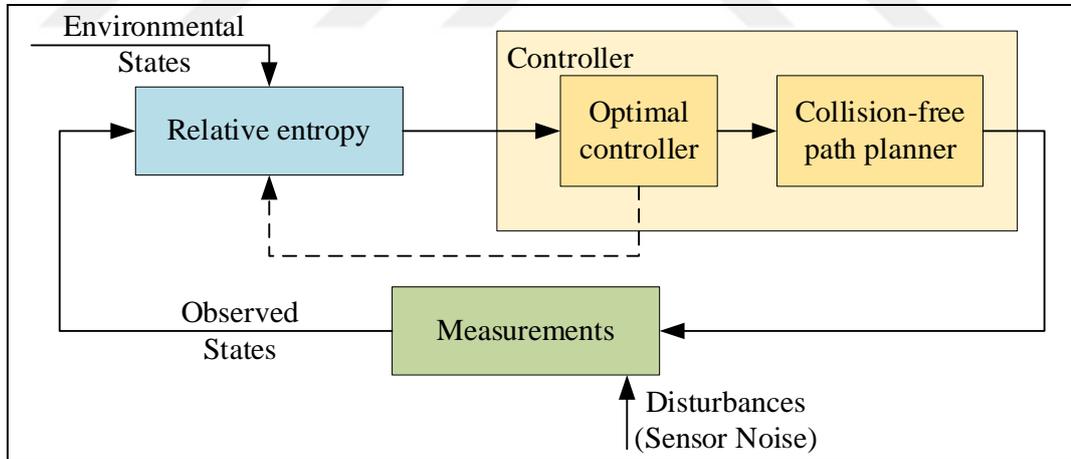


Figure 4.1 : Informative path planning block diagram.

Detailed explanations of relative entropy and information seeking optimal control topics are presented in the following subsections.

4.1 Relative Entropy

Relative entropy, also called Kullback-Leibler Divergence (KLD), is a measure of probability difference between two random variables. This measure also helps to find

a correlation between these variables with mutual information definitions. The uncertainties of the observations are defined by the probabilistic approaches given in previous sections. The uncertainties of the observations and the desired level of uncertainty of the environmental states determine the entropy difference between those variables, as shown in Figure 4.2. Note that, the entropies are calculated through the probabilities of the variables. As a result, the KLD provides a framework that can be used with information-theoretic and probabilistic approaches with given entropy definitions.

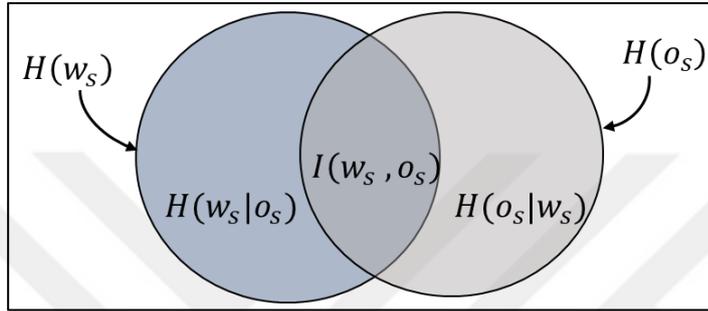


Figure 4.2 : Venn scheme representation of the KLD. The entropy of the environmental states, $H(w_s)$, the entropy of observations, $H(o_s)$, conditional entropies between environmental states and observed states, $H(w_s|o_s)$, vice-versa, $H(o_s|w_s)$ and the mutual information between environmental states and observed states, $I(w_s, o_s)$.

In other words, uncertainty measures between two random variables can be expressed with its formulation given below (Kullback & Leibler, 1951) ;

$$d_{KL}(P(w_s)||P(o_s)) = \int P(w_s) \log \left(\frac{P(w_s)}{P(o_s)} \right) \quad (4.1)$$

Here, $P(w_s)$ is the occupancy probability of volumetric space, s (Figure 4.4) in the whole map of the specified area, M_w . Similar to this, $P(o_s)$ is the occupancy probability of s in the observed map of the specified area M_o . Moreover, the mutual information between observations and environment is as following with Shannon's Entropy definition (Shannon, 1948);

$$I(w_s, o_s) = H(w_s) - H(w_s|o_s) \quad (4.2)$$

Where, $H(w_s)$ is the entropy of the environmental states, and $H(w_s|o_s)$ is the conditional entropy of environmental states and observed states. With the mathematical operations mutual information with the following KLD expressions can be written as below;

$$I(w_s, o_s) = d_{KL}(P(w_s, o_s)||P(w_s)P(o_s)) \quad (4.3)$$

Here, $P(w_s, o_s)$ is the joint probability of w_s and o_s random variables.

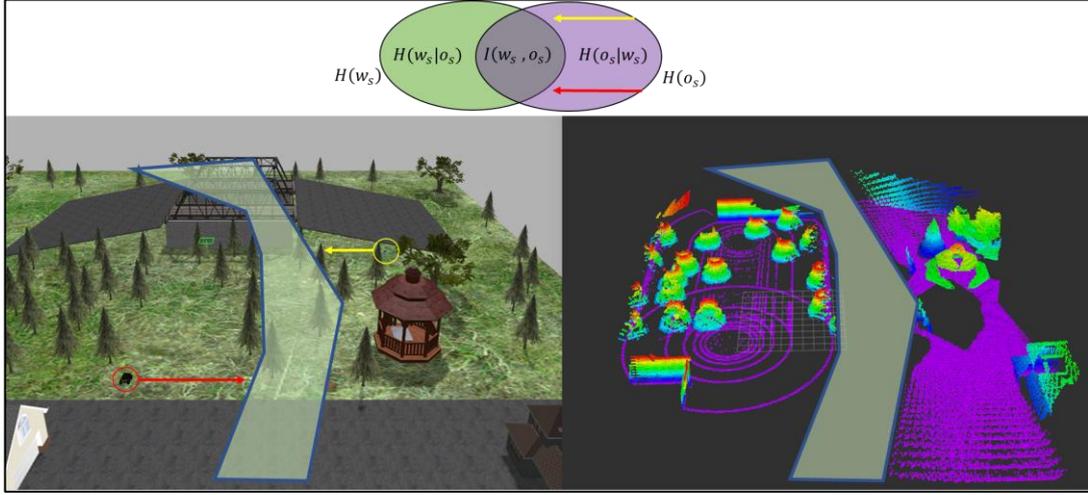


Figure 4.3 : Representation of conditional entropy of states. The transparent green area denotes the uncertain area of the environment. UAV and UGV must move along with the yellow and red arrows, respectively. With the movement of robots, the information, $I(w_s, o_s)$ will increase, and conditional entropy will decrease.

So, maximizing the KLD of $P(w_s, o_s)$ over $P(w_s)P(o_s)$ results, maximizing the mutual information (or minimizing uncertainties) between robots' observations and environmental observable states, as in Figure 4.3. Note that, maximizing $P(w_s, o_s)$ over $P(w_s)P(o_s)$ is not the same expression with maximizing KLD of $P(w_s)$ over $P(o_s)$. If the process of w_s and o_s is independent, the $P(w_s, o_s)$ will be equal to $P(w_s)P(o_s)$. As a result, KLD between these two processes are maximized, however, $I(w_s, o_s)$ will be equal to zero. To sum up, the independent variables do not have any mutual information. Maximization of the mutual information between the environmental state and the robots' observations is one of the aims of this thesis. Information maximization or seeking controller must be determined to serve the purpose of this dissertation.

4.2 Information Seeking Optimal Control

The planning strategy will be handled as an information seeking controller, which maximizes the time-varying local and global states' negative joint posterior. HeRT members receive their control inputs in order to provide maximized information on the environment. These control inputs will ensure the UGV and UAV heading towards to the volumetric space, s where the number of permanently occupied voxels are under the threshold of mutual information. Note that, this threshold is not the same with

OctoMap occupancy probability upper and lower bound. However, these bounds will determine the exploration effectiveness in accordance with the mission duration and obtaining a better map. In order to provide these outcomes with an information seeking controller, an objective function, which maximizes the mutual information, must be defined.

$$O(u^r(t)) = \operatorname{argmax}(d_{KL}(P(w_s(0:t_f), o_s(t+1)) || P(w_s(0:t_f)) P(o_s(t+1)))) \quad (4.4)$$

With;

$$\begin{aligned} & d_{KL}(P(w_s(0:t_f), o_s(t+1)) || P(w_s(0:t_f)) P(o_s(t+1))) \\ &= \iint P(w_s(0:t_f), o_s(t+1)) \log \frac{P(w_s(0:t_f), o_s(t+1))}{P(w_s(0:t_f)) P(o_s(t+1))} \end{aligned} \quad (4.5)$$

Where, $u^r(t)$ is the control input of a robot at time t and $P(w_s(0:t_f))$ refers reachable occupancy probability of s from the beginning to end of the mission in the whole map of the specified area, M_w . It is determined with regard to permanently occupied voxels in each volumetric space on the obtained in OctoMap. $P(o_s(t+1))$ is the occupancy probability of s with the measurement of robots at the time $(t+1)$ where the robot changes its states;

$$X(t+1) = f(X(t), u^r(t)) \quad (4.6)$$

Here, the state includes the points of interest (target point), $p_c(x_{p_c}, y_{p_c}, z_{p_c})$. HeRT member changes its state to move to the volumetric space's center point, p_c in order to maximize the information about this space. The projection of that center point on the X-Y plane is the target point of UGV, $p_{c_{ugv}}$. For UAV, the target point, $p_{c_{uav}}$ is determined as the intersecting point of the X-Y plane's normal that includes the center point and the X^{uav} - Y^{uav} plane. The X-Y plane at UAV's flight altitude is denoted as X^{uav} - Y^{uav} plane. So, moving to the target point affects $P(o_s(t+1))$, and converging $P(o_s(0:t_f))$ to $P(w_s(0:t_f))$ is desired. Also, energy consumption must be taken into account in order to define the next target point for each robot. For the sake of simplicity, energy consumption is implemented to the objective function as a weight value calculated with the formulation given below;

$$\omega_i = \frac{1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \quad (4.7)$$

Here, ω_i is a strictly positive scalar value, i is the number of target points, x_2 and y_2 are the target point values in x and y coordinate, respectively, at the time t . x_1 and y_1 are the target point values visited at time $t - 1$.

To get an optimal control input for a single robot, the gradient of the objective function with respect to its target point state can be written as follows;

$$\frac{\partial O_k}{\partial p_{c_i}} = \iint \frac{\partial P(w_s|o_s)}{\partial p_{c_i}} P(o_s) dw_s do_s + \iint \frac{\partial P(o_s|w_s)}{\partial p_{c_i}} P(w_s) \log \frac{P(w_s|o_s)}{P(w_s)} dw_s do_s \quad (4.8)$$

With the mathematical operations;

$$\frac{\partial O_k}{\partial p_{c_i}} = \iint \frac{\partial P(o_s|w_s)}{\partial p_{c_i}} P(w_s) \log \frac{P(w_s|o_s)}{P(w_s)} dw_s do_s \quad (4.9)$$

Proof of attaining equation (4.9) from equation (4.8) is given in Appendix C.

So, with the ω_i implemented to the control input, the controller of the robot will be given as below;

$$u_t^r = \omega_i \frac{\partial O_k}{\partial p_{c_i}} \quad (4.10)$$

Theorem 4.1. *Equation (4.10) with the movement weight will converge to zero between obtained measurements by all HeRT members. The measurement probabilities differ from each other at each point with respect to OctoMap upper and lower bounds, and equals to zero outside the robot's observation area. Also, equation (4.15) is Lyapunov stable subject to its local optimality in order to maximize the objective function.*

Proof of Theorem 4.1. *This proof is acquired with the jointly use of theorems given by Julian et al. (2012), Palomar & Verdú (2007) and the approach given in this dissertation with target point gradient and positive scalar weight value:* In order to check the stability of the system with the control inputs, Lyapunov function candidate, $V_k = -O_k$ is presented. Within the use of equation (4.10), the closed-loop system dynamics can be written as;

$$\frac{dp_{c_i}}{dt} = -\omega_i \frac{\partial V_k}{\partial p_{c_i}} \quad (4.11)$$

If $\partial P(o_s|w_s)/\partial p_{c_i}$ is continuous, $\partial V_k/\partial p_{c_i}$ will be continuous on the space $p_{c_i} \in \mathbb{P}$. As a result, equation (4.11) is locally Lipschitz with a continuously differentiable

candidate function. The Lie derivative of the Lyapunov candidate function, which is definitely negative or equal to zero.

$$\sum_{i=1}^S \frac{\partial V_k}{\partial p_{c_i}} \frac{dp_{c_i}}{dt} = - \sum_{i=1}^S \omega_i \left(\frac{\partial V_k}{\partial p_{c_i}} \right)^2 \quad (4.12)$$

So, the partially derivative of the conditional entropy of o_s over w_s in respect of p_{c_i} can be written as follows with the assumption of the area around the target point is not visited before;

$$\frac{\partial P(o_s|w_s)}{\partial p_{c_i}} = 0 \quad (4.13)$$

Therefore, the system is bounded under all solutions. Regarding the invariant equilibrium points p'_{c_i} which are providing local optimality of V_k ;

$$\left. \frac{\partial V_k}{\partial p_{c_i}} \right|_{p_{c_i}=p'_{c_i}} = 0 \quad (4.14)$$

means $\partial O_k / \partial p_{c_i} = 0$, with $i = \{1, \dots, S\}$, and S is the total number of volumetric space.

As a result, targets of the robots will converge to p'_{c_i} considering LaSalle's Invariance principle, all the requirements are fulfilled. Moreover, all target points are Lyapunov stable subject to its locally optimal case with respect to the objective function's maximization.

$$\left. \frac{\partial O_k}{\partial p_{c_i}} \right|_{p_{c_i}=p'_{c_i}} = 0 \quad (4.15)$$

and p'_{c_i} is the equilibrium points to provide local optimality. Moreover, these points are chosen from a set of possible volumetric spaces in the environment, $p'_{c_i} \in P_c, \{1, \dots, S\}$. □□□

Rule 1 These target points will be sent to the robots with the conditions given below with the $p_{c_i}^z$ and h_2 which refers to the coordinate point value in the z-axis and the height value of the layer given in Section 3, respectively. In other words, if $a + a/2 < h_2$ in Figure 4.4, $p_{c_{i+1}}$ will be assigned to UGV as $p_{c_{ugv_{i+1}}}$.

$$p_{c_i} = \begin{cases} p_{c_i}^z < h_2, & p_{c_{ugv_i}} \\ p_{c_i}^z > h_2, & p_{c_{uav_i}} \end{cases} \quad (4.16)$$

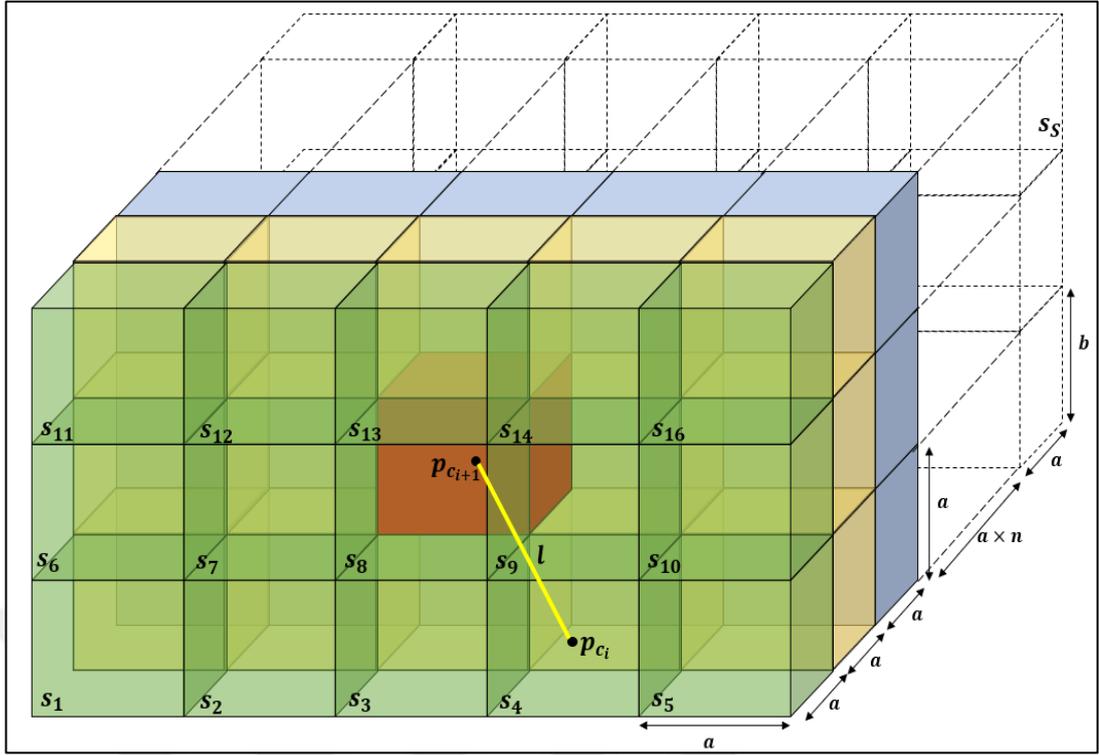


Figure 4.4 : Representative illustration of volumetric spaces. In this illustration; $l = 1/\omega_i$. Green transparent boxes (s_1 to s_{16}) are the ones that mutual information value is over the threshold, $I(w_{s_1:s_{16}}, o_{s_1:s_{16}}) > I_{th}$. The red box is the volumetric space whom mutual information value is under the threshold, $I(w_{s_{24}}, o_{s_{24}}) < I_{th}$. Dimensions of the volumetric space, s is $a \times a \times b$ and $p_{c_{i+1}}$ is the next target point to visit.

Rule 2 UGV create their trajectories to reach the target point with the obstacle avoidance constraints given in Appendix A. However, if $p_{c_{ugv_i}}$ is unreachable because of the obstacles, the target point will transform into a circle with a diameter d_i , and the value of the diameter will be $0 < d_i \leq D_{tolerance} \in R$. In addition, if $p_{c_{ugv_i}} + D_{tolerance}/2$ is also unreachable by UGV, and this target point will be assigned to UAV.

Rule 3 UGV and UAV has a certain knowledge of its localization in the environment, bounds of the environment and the joint measurement probabilities, $P(o_s | w_s)$.

The flow of the information seeking optimal control process is given in Figure 4.5. At time $t - 1$, the probability of conditional states between the environment and each robot is approximated. The occupancy probabilities of the volumetric spaces are calculated with the use of UGV's and UAV's measurements at time t . After that, the control inputs, u_t^{uav} and u_t^{ugv} , calculated in order to move robots to the target points

depending on decreasing uncertainties about the environment. Finally, at the target point $p_{c_{i+1}}$, observations and the probability of conditional states $P(w_s|o_s)$ will be updated.

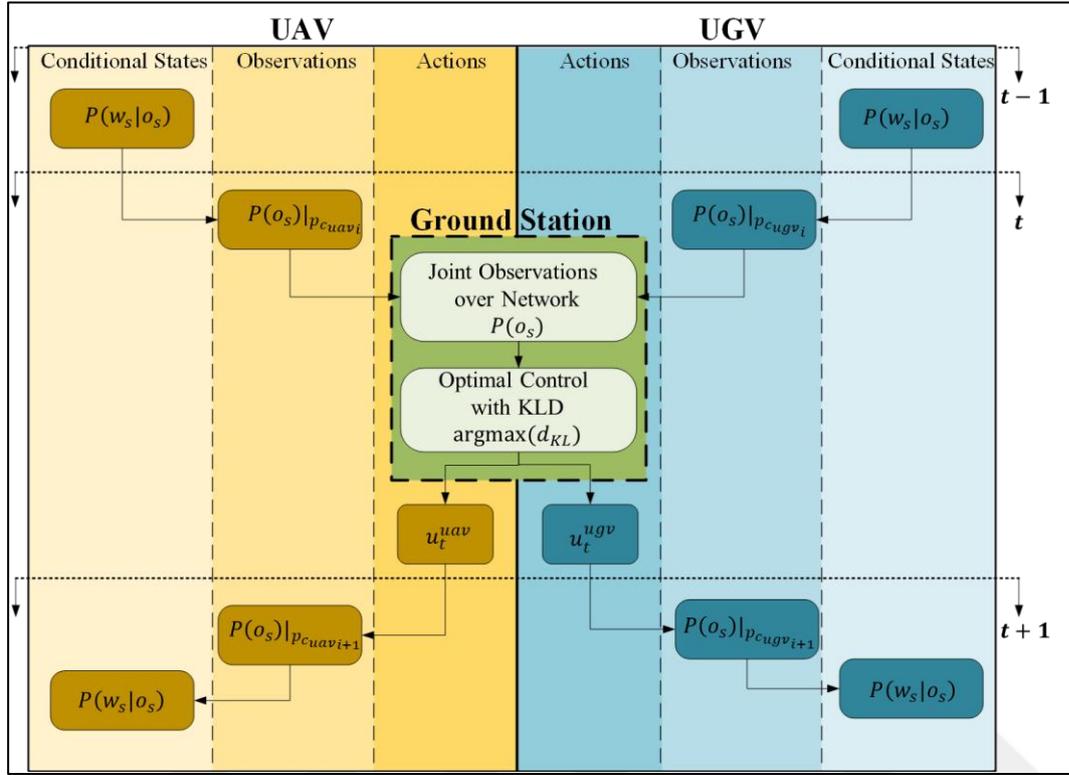


Figure 4.5 : Flow diagram of informative path planning in this dissertation. This flow diagram includes phases, such as; before $t - 1$, from $t - 1$ to t and from t to $t + 1$. Also, it shows the process flow from the conditional states to the observations and actions of UGV and UAV with the ground station.

In summary, informative path planning for HeRT with relative entropy theory and an optimal controller in association with KLD is explained. As a result, a Lyapunov stable optimal controller is achieved through optimization of objective function defined by KLD and weight as stated by the Theorem 4.1. In order to verify the given theoretic approaches, the case studies are handled in the next section.

5. CASE STUDIES

The theoretical background of this study given in previous sections has also experimented for the purpose of verifying the solution method in the simulation environment and the real world.

Experimental studies are only done for determining the similarity metric for constructing common maps for the HeRT. Unfortunately, because of security issues about flying UAVs without pilot in-the-loop is forbidden around public areas in Turkey. As a result, active SLAM with informative path planning strategy for HeRT is only validated in the simulation environment.

5.1 Experimental Studies

Experiments are handled in order to determine the suitable similarity metric for constructing common maps for HeRT. The experimental setup, experiment area, and results are given below.

5.1.1 Hardware Setup

The aim of the experimental studies is the observation of the theoretical algorithms in the real world. These studies are realized with the equipment listed in Table 5.1.

Table 5.1 : Experimental study equipment.

System Equipment	Intended Use
MATRICE 600PRO	Unmanned Aerial Vehicle
HUSKY A200	Unmanned Ground Vehicle
D-RTK GPS	Obtaining Location information with sub-centimeter accuracy on UAV
IMU-1	Internal inertial measurement unit on UAV
IMU-2 (Xsens)	External inertial measurement unit on UGV
Modem	Receive information on UGV
GPS	Obtaining Location information with on UGV
Rugged Computer	Processing data on UGV
LIDAR VLP-16	Laser sensor on UGV
LIDAR VLP-16Lite	Laser sensor on UAV
Mounting Parts-1	To assembly sensors on UGV and UAV
Mounting Parts-2	To assembly sensors on
Manifold	The onboard computer in order to process data on UAV
Main Computer	Receiving and processing data coming from UAV and UGV

These equipment are the main components of the experiments and the detailed information in Appendix D. An integrated experimental setup is provided with these components specific for this dissertation.

5.1.1.1 Integrated experimental setup

The experimental setup includes three main systems, such as; UGV system, UAV System and Ground Station.

UGV and UAV systems are used for obtaining data respectively from the ground and the air. The ground station was used for collecting data transferred from UGV and UAV systems. Also, it was used to processing these data and transmitting control inputs to systems. In Figure 5.1, experimental setup contents are shown. It can be seen that UGV, UAV and Ground Station are used. Detailed information about these systems is given below.

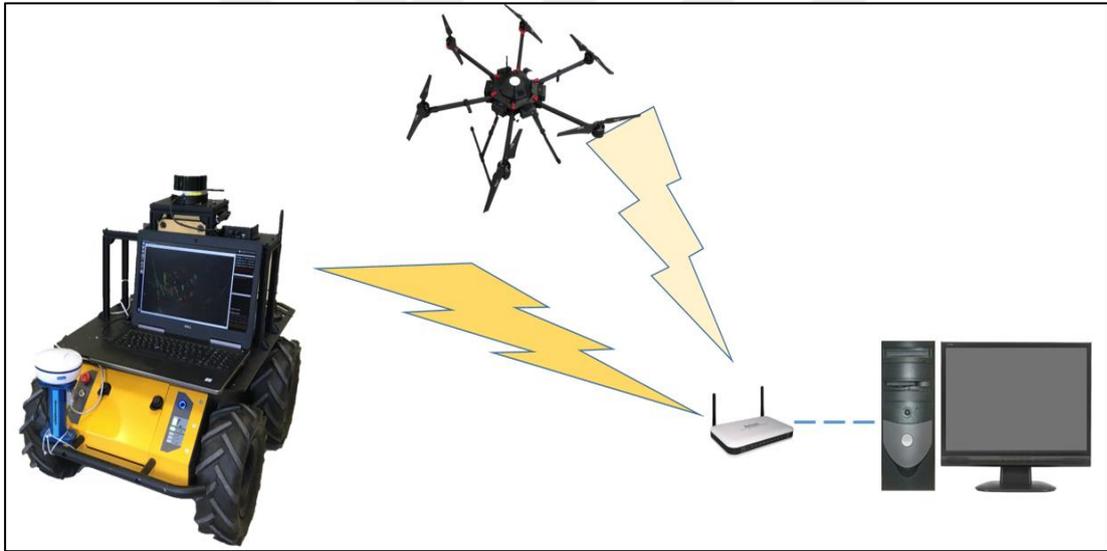


Figure 5.1 : Experimental setup principle of operation.

Data rates of the hardware used in the experiments of this study are given in Table 5.2.

Table 5.2 : Data rates of the hardware.

Hardware	Data Rate (Hz)
Velodyne VLP 16 and VLP 16Lite	10
DJI IMU	400
D-RTK GPS	10
UGV GPS	10
Xsense IMU	100
UGV Encoder	3

UGV system

UGV experimental setup includes HUSKY A200, rugged computer, VLP-16, VLP-16 interface box, GPS sensor, external Xsens IMU sensor, and mounting parts can be seen in Figure 5.2



Figure 5.2 : UGV experimental setup-outside.

Ground Station

The ground station is employed to control UGV and UAV and receive data from vehicles remotely. It has a high-performance processor unit and a graphic card suitable for missions of this dissertation. A wireless link handles the data transmission process. The ground station has Intel Core i7-6820HQ CPU 2.70GHz processor, and 32 GB installed memory (RAM).

UAV System

UAV system contents MATRICE 600PRO, D-RTK GPS, VLP16-Lite Lidar, VLP16 Interface Box, manifold and mounting parts.

5.1.2 Software Setup

The data flow of the hardware and the software in the experimental setup is given in **Figure 5.3**. Robot Operating System (ROS) -Indigo release is used as the base of the system software. EKF localization¹, Navsat Converter, LOAM Velodyne and OctoMap packages are utilized with customized settings for the specific experiments in this dissertation. Customized settings are determined with regard to hardware specifications, such as; measurement errors, process errors and physical attributes. Also, map resolution, occupancy state determinant bounds and surface roughness

¹ EKF localization is only utilized for error calculation in experimental studies.

parameters are included in order to obtain valid results considering the theoretical background of the thesis given in previous sections.

In addition, the software is developed in Python language for height mapping, layering, similarity calculations and map merging algorithms. Nodes, message types, broadcasting transforms, publishing and subscribing topics with the defined queue size are also written based on Python language.

In conclusion, the desired outputs are obtained on the software side with the hardware drivers, customized ROS packages and developed software.

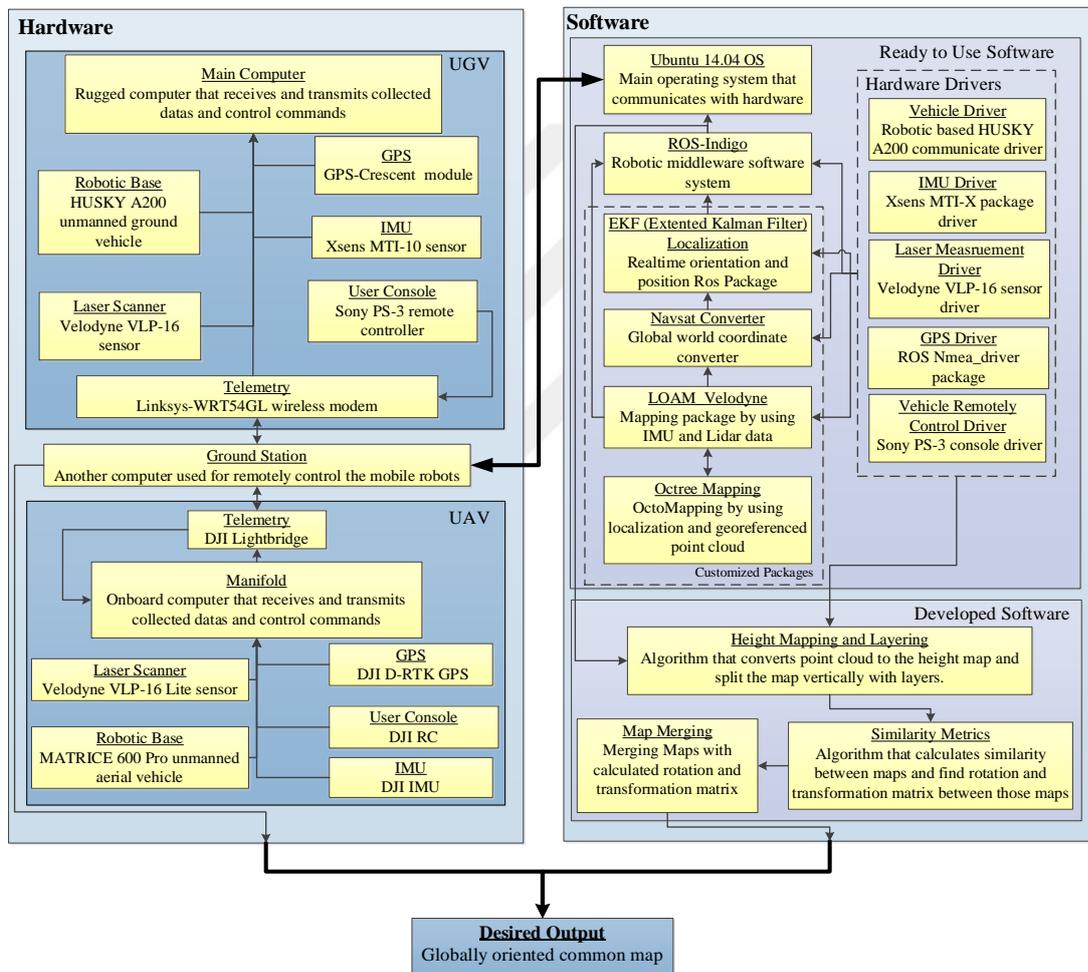


Figure 5.3 : Experimental setup physical and functional block diagram.

5.1.3 Experiment Area

Experiments were carried out on Istanbul Technical University Ayazaga Campus. The field within the campus, chosen for the experiments, has different types of structures, slopes, and objects which can provide the main idea to handle common maps. Trees

and roofs are essential for this study, which causes difficulties in sensing, as stated in Figure 2.3. Images of the experiment area can be seen in Figure 5.2 and Figure 5.4.



Figure 5.4 : Experiment area photo was taken by UAV.

5.1.4 Scenario

Experiments were performed within the created scenario that includes hardware setup, software structure and sorted processes. Hardware setup and software structure were given in previous subsections. The data collection method with UAV and UGV, and calibration of VLP-16 and VLP-16Lite were the same as in (Aybakan et al., 2019). However, post-processing of the data differs from the given study.

In order to mention scenario, the sorted processes can be given as below;

- 1- Prepare, and power on UGV and UAV.
- 2- Define the paths of UGV and UAV in order to cover the experiment area as in (Aybakan et al., 2019).
- 3- Fly UAV autonomously with the given trajectories.
- 4- Collect the data from UAV, such as; point cloud, $m_{t_{uav}}$, RTK GPS signals and IMU, and send it to the ground station via the wireless link.
- 5- Power off the UAV system.
- 6- Operate UGV manually with the given trajectories.
- 7- Collect the data from UGV, point cloud, $m_{t_{ugv}}$, and send it to the ground station via the wireless link.
- 8- Power off the UGV system.

9- Process the data with the following algorithm;

Algorithm 1 Constructing common map of the experiment area

Input: $C_{(n,i)t-1uav}^L, C_{(n,i)t-1ugv}^L$
Output: $M_{wc}, A_m, G_m, \tilde{u}, V_i$
begin Collect points $C_{(n,i)t_{uav}}^L$ and $C_{(n,i)t_{ugv}}^L$ from the optical channel $S_{n_{uav}}$ and $S_{n_{ugv}}$

```

for  $t = 1: \delta_z: t_f$  do
    Calculate surface roughness,  $\kappa_t$  for each time step,  $\delta_z = 0.1sec$  for UAV and UGV
    if  $\kappa_t < \kappa_{th} = 0.1$ ;
         $C_{(n,i)}^L = \Gamma_n$ 
        then; Compute to line distance  $d_\Gamma$  and obtain  $f_\Gamma(C_{(n,j)_t}^L, T_{\sigma,i}^L)$ 
        else;
             $C_{(n,i)}^L = A_n$ 
            then; Compute the distance to the line  $d_\Delta$  and obtain  $f_\Delta(C_{(n,i)_t}^L, T_{\sigma,i}^L)$ 
        end
        Calculate  $f(T_{\sigma,i}^L) = d$ 
        Calculate the position difference  $T_\sigma^L = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]^T$  at the time,  $t$ 
        Calculate translation,  $T_{\sigma,i}^L$  and rotation,  $R$  for UAV and UGV based on
        nonlinear optimization given in equation (2.14)
        if  $\min(T_{\sigma,i}^L - (J^T J + \lambda \text{diag}(J^T J))J^T d)$  converges
            then; Break
        end
        Obtain georeferenced point cloud for UGV and UAV, and measurements for each
        time step;  $M_{LOAM_t}, m_{t_{ugv}}$  and  $m_{t_{uav}}$  respectively
        Calculate  $l_{UB_o}$  and  $l_{LB_o}$  with regard to LOAM error, 4.5% by Equation 9
        for  $j = 1: 4$  and  $l_{UB_o} = 3.1, l_{LB_o} = -2.2$  do
            for  $i = 1: 12^6$  do
                Calculate  $l(V_i | m_{1:t})_j$  with  $l_{UB_o}$  and  $l_{LB_o}$ 
                Obtain local OctoMap of UGV and UAV for each layer,  $V_{ugv_j}$  and  $V_{uav_j}$ ,
                respectively
            end
            for  $i = 1: 30^3$  and  $l_{LB_h} = -2, l_{UB_h} = 3$  do
                Calculate  $l(g_i | m_{1:t})_j$  for each layer  $j$  with
                if  $P(g_i | m_t)_j > 0.3$ ;
                     $z_{t_j} = z_{i_j}$ 
                else;
                     $z_{t_j} = 0$ 
                end
                Obtain local height map of UGV and UAV for each layer,  $H_{0:t_{ugv}_j}$  and  $H_{0:t_{uav}_j}$ ,
                respectively
            end
            #Initial value of  $u$ , with  $15^\circ$  rotation and  $[x \ y]^T = [1m, 1m]^T$ , translation difference
             $u = \begin{bmatrix} 0.96 & 0.26 & 1 \\ -0.26 & 0.96 & 1 \\ 0 & 0 & 1 \end{bmatrix}$ 
            for  $i = i + 1$  do
                Calculate distances or entropies of the heights for each layer with the
                equations in Section 3.2
                 $\tilde{u}_j = \text{argmin } H$  or  $\tilde{u}_j = \text{argmin } d$ 
                if  $\tilde{u}_j$  converges
                    then; Break
                end
            end
        end
        Calculate  $G_m(z)$  and  $A_m(z)$  with  $\tilde{u}$ .
         $M_{wc} = G_m + A_m$ 
    end
end

```

Note that, only four layers are employed for this study, and the 4th layer has the information between 925mm to 40.000mm.

5.1.5 Results

Construction of common height maps with seven different entropy-based similarity metrics and utilization of the layering method in this study is realized with experiments. Both ground and aerial robots collected useful data in order to examine the environment with the presented method in this thesis. Aerial and ground maps are merged, and each point cloud is moved to the desired global coordinate system's origin (globally-oriented), designated in this study as the UAV's local coordinate system's origin. As a result of the study, mapping of the environment with complex features is realized. The details of how metrics are computed are stated below;

The number of spatial samples is 90.000 because of the 300×300 grid construction. The number of histogram bins used in the joint histogram computation is 100 – due to the 40cm layers of 40m total examine height.

Results are investigated in two different parameters; root mean square error (RMSE) of heights in merged maps and computation time; also, visual outputs are given.

5.1.5.1 Height Value RMSE

Since UAV has the RTK GPS onboard, aerial height map obtained by EKF localization can be the reference to calculate RMSE of heights in the merged map. Therefore, the root RMSE formulation can be;

$$rmse = \sqrt{\left(\frac{(\text{sum}(R(A_m(z)) + T - M_m(z))^2)}{n}\right)} \quad (5.1)$$

Where $A_m(z)$ and $M_m(z)$ are respectively, the aerial and referenced height map of the environment. R is the rotation matrix, T is the translation matrix. Sum refers to the height values summation. n is the number of grids on the map, which is 300×300=90.000.

In Figure 5.5, results are inserted into a chart for the purpose of understanding the relation between RMSE values and computation times of different similarity metrics with the layering method and without the layering method. As can be seen from the chart, more complicated similarity metrics provide less error at the cost of increased

computation time. Moreover, the layering method provides, on average %46 less error, with an average 0.1sec performance improvement on the computation time. Jensen Divergence is determined as the best similarity metric in this study with 0.92m RMSE value. Without the layering method, RMSE values increase for each similarity metric because of the experiment area's characteristics explained in previous sections. The more features (heights) are included for the calculation of the similarity metric, the more accuracy is obtained.

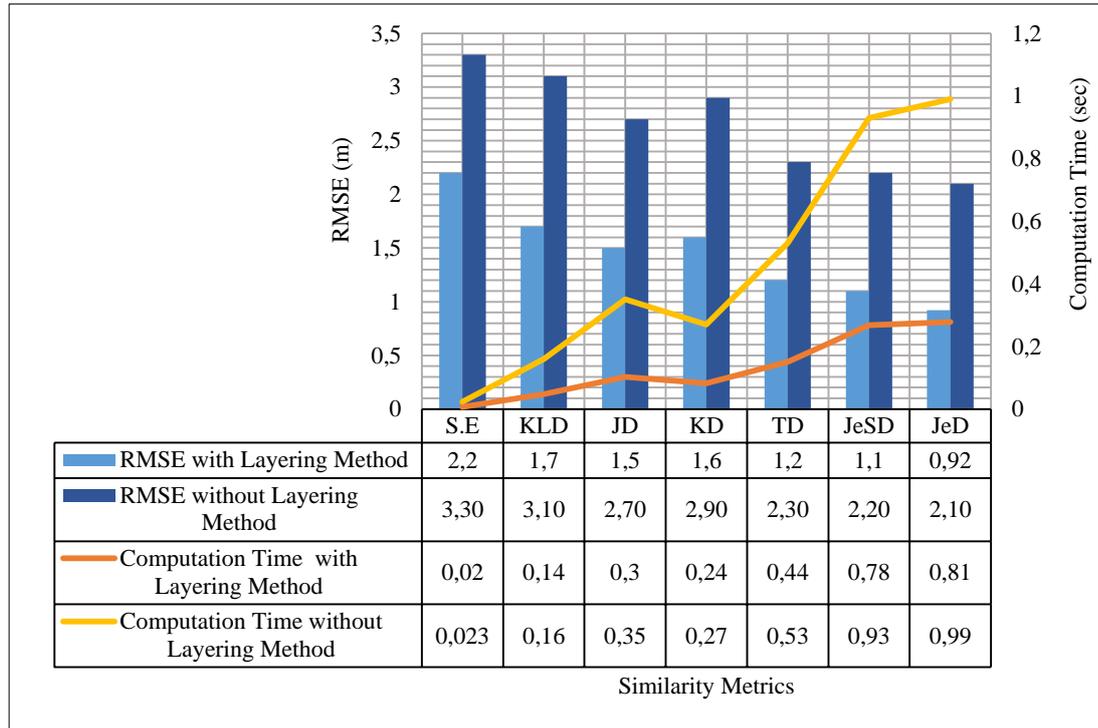


Figure 5.5 : Chart of similarity metric versus computation time and RMSE.

Also, it is expected that the computation time will increase to determine the best similarity with the layering method because four height maps are included in the calculation instead of one map. However, the number of iterations stated in Table 5.3 substantially rise in order to find the best position and orientation, \tilde{u} . This result shows that, \tilde{u} is found with fewer iterations for each layer than the complete height map without layers. When calculating the \tilde{u} , the results converged to the same values after the iterations stated in Table 5.3.

Table 5.3 : Number of iterations to calculate the best position and orientation with and without the layering method for different similarity metrics.

Similarity Metric		SE	KLD	JD	KD	TD	JeSD	JeD
Number of Iterations	With Layering Method	8	46	88	75	105	172	162
	Without Layering Method	10	51	12	84	127	200	212

TD and JeSD metrics had similar RMSE and adjacent values (Figure 5.6) as their metric formulations are also similar. JD has better RMSE value than KD even though the JD is the symmetric version of the KLD, and KLD has the formulation as half of KD. It occurs because Jeffreys divergence similarity calculation relies on the elevation values of both vehicles more than KLD and KD metrics.

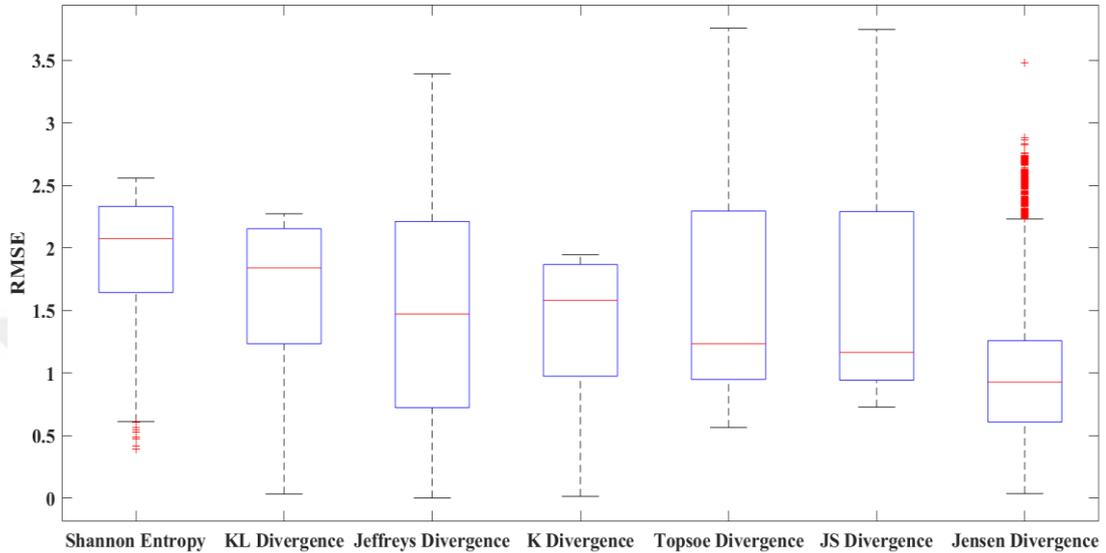


Figure 5.6 : Box plot of the similarity metrics according to RMSE values.

As a result, less than 1m RMSE value provides a high level of confidence for the obtained common map, where the maximum height value is 40m. Minimum error values close to zero are obtained in grids that are important for the ground robot's movement to climb over. It achieves better results for lower height values in the first two layers. A maximum RMSE value of less than 2.5m and outliers between 2m and 3m are negligible since those grids can be set as insignificant to the operation of agents. Even though the calculation time is higher than the others, JeD is determined as the best similarity metric for this specific dissertation.

5.1.5.2 Visual Outputs

Visual outputs of the experimental studies are given as; height maps of the layers for each robot, merged OctoMap, merged height map and the merged point cloud map of the environment.

The obtained height maps' 2D views of the layer-1, layer-2 and layer-4 for UGV and UAV can be seen in Figure 5.7. Similar to Figure 3.3, different colours are assigned for each layer to understand the similarities with ease. For the first layer; blue refers to the heights of the objects and magenta refers to the base of the investigated area. Aqua

colour is chosen for the second layer, and green colour refers to the third layer. Finally, for the fourth layer, yellow is selected.

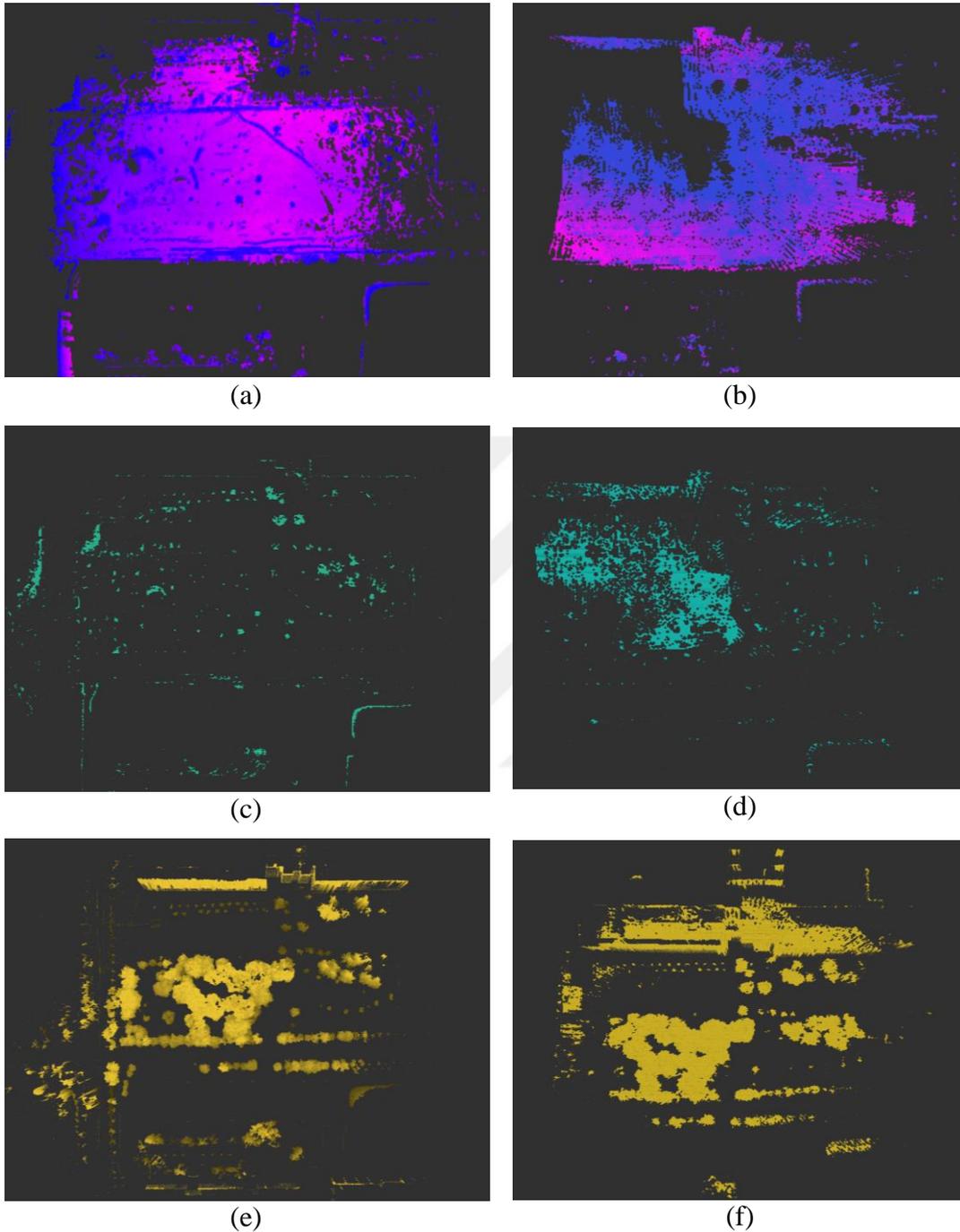
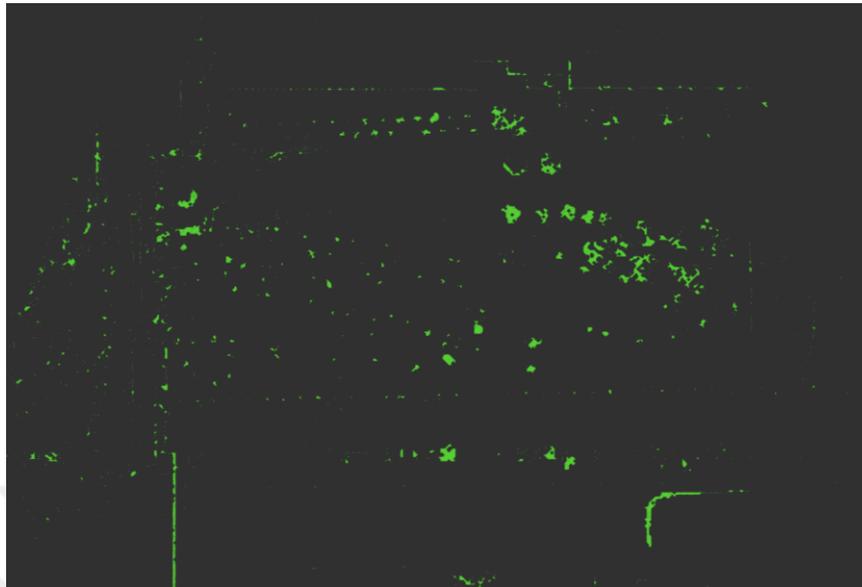


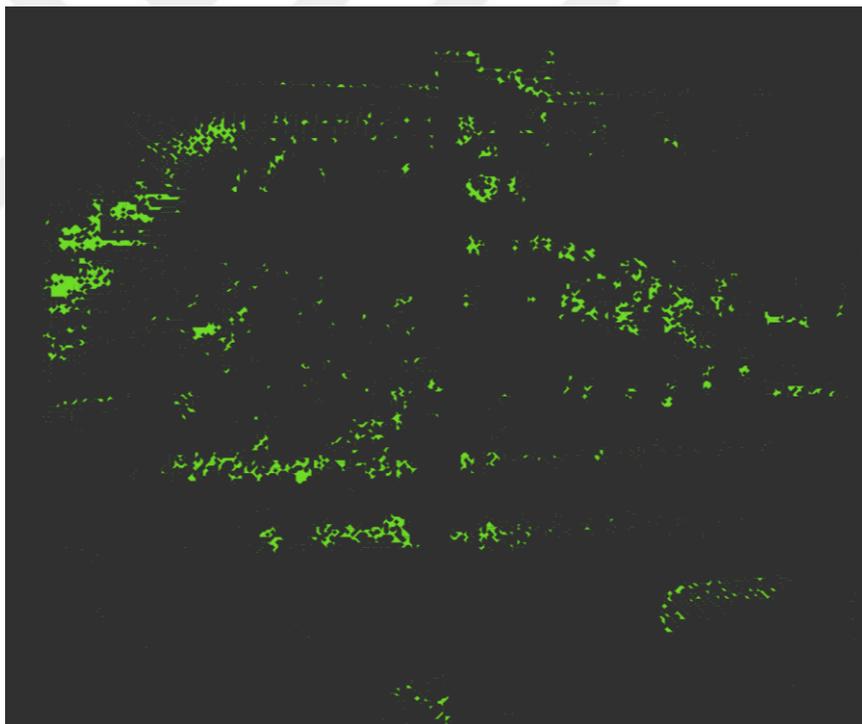
Figure 5.7 : Obtained height maps of the layers. (a) UGV layer-1. (b) UAV layer-1. (c) UGV layer-2. (d) UAV layer-2. (e) UGV layer-4. (f) UAV layer-4.

As can be seen from Figure 5.7, height maps of the layer-1 and layer-2 differ from each other. The height maps of the layer-4 seem similar by visual inspection; however, the best similarity is found between the height maps of layer-3, which is given in

Figure 5.8. This result is compatible with the visual outputs, and it is reasonable because the third layer is the most common layer scanned by UGV and UAV.



(a)



(b)

Figure 5.8 : Obtained height maps of the layers. (a) UGV layer-3. (b) UAV layer-3.

Figure 5.8 shows that; most of the heights on layer-3, which are observed by UGV and UAV, are similar. Note that; in Figure 5.7 and Figure 5.8, realized images are aligned, despite that these are not referenced to a coordinate system. Images are arranged in order to quote them from the same angle of view. Actual layer images of UGV and

UAV have position and orientation differences before calculating the similarities of heights and applying the transformations.

The common elevation map obtained by the experiments can be seen in Figure 5.9

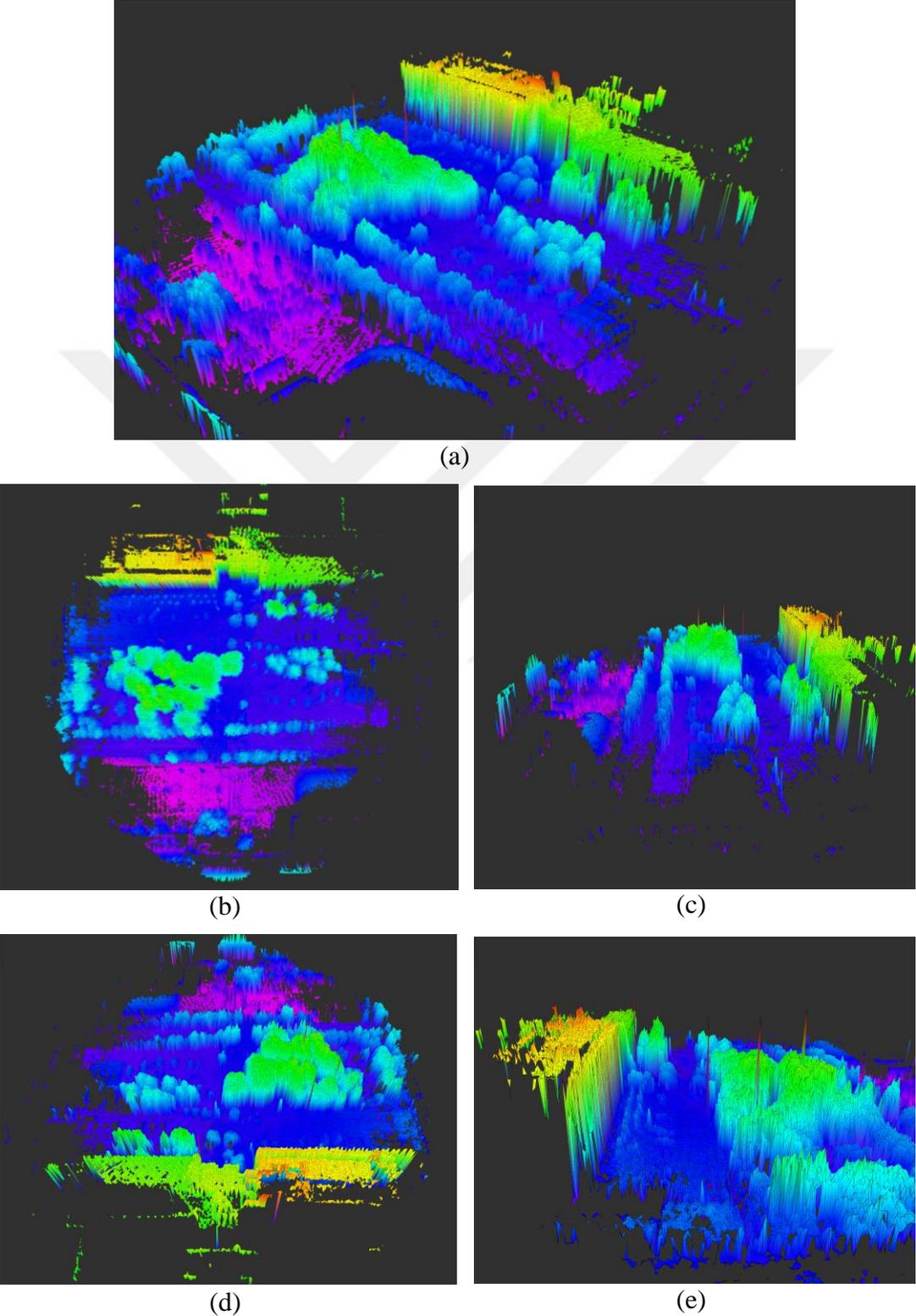
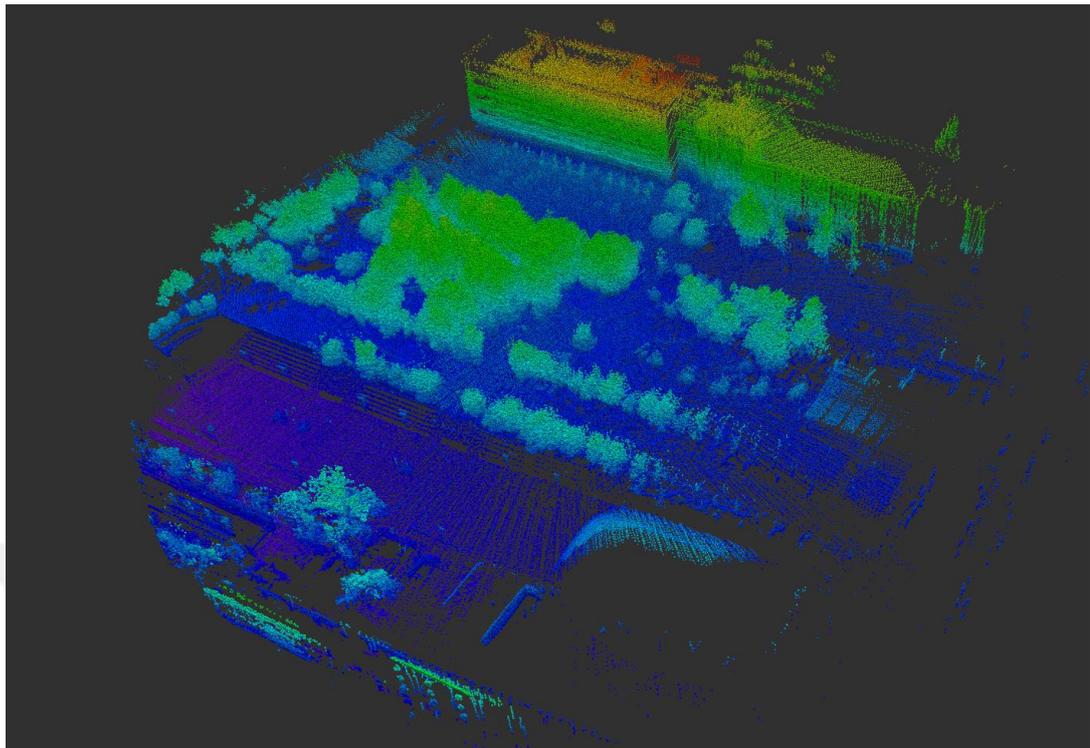
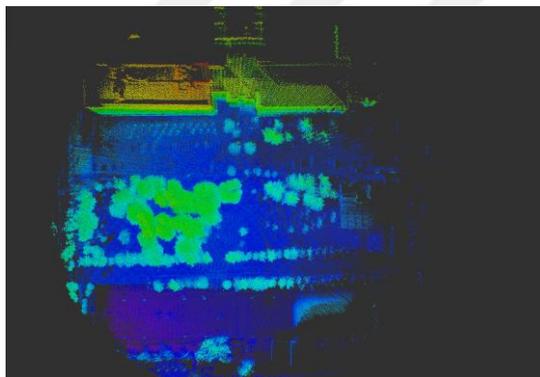


Figure 5.9 : Obtained common height map of the environment. (a) to (e) various views of the height map.

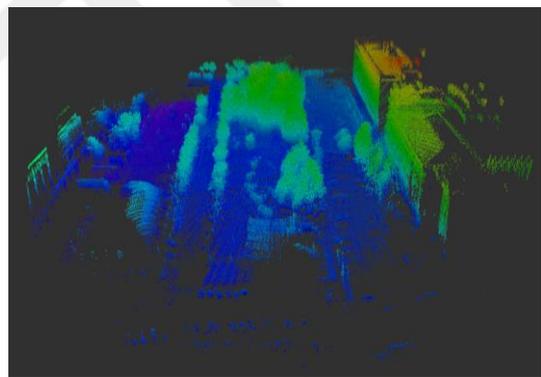
The common OctoMap obtained by the experiments can be seen in Figure 5.10



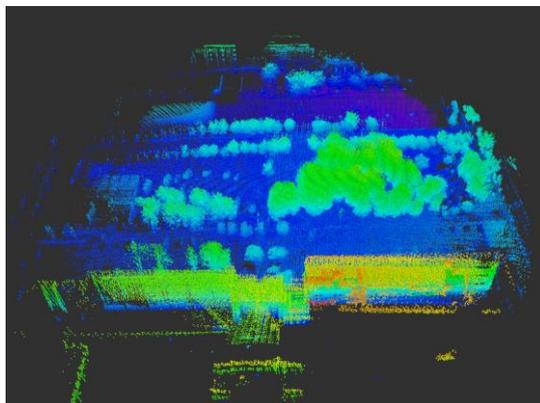
(a)



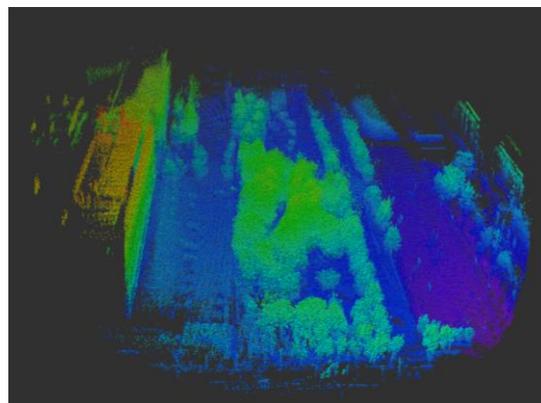
(b)



(c)



(d)



(e)

Figure 5.10 : Obtained OctoMap of the environment. (a) to (e) various views of the OctoMap with resolution=0.2m.

The achieved common point cloud map obtained by the best similarity metric (JeD) transform can be seen in Figure 5.11.

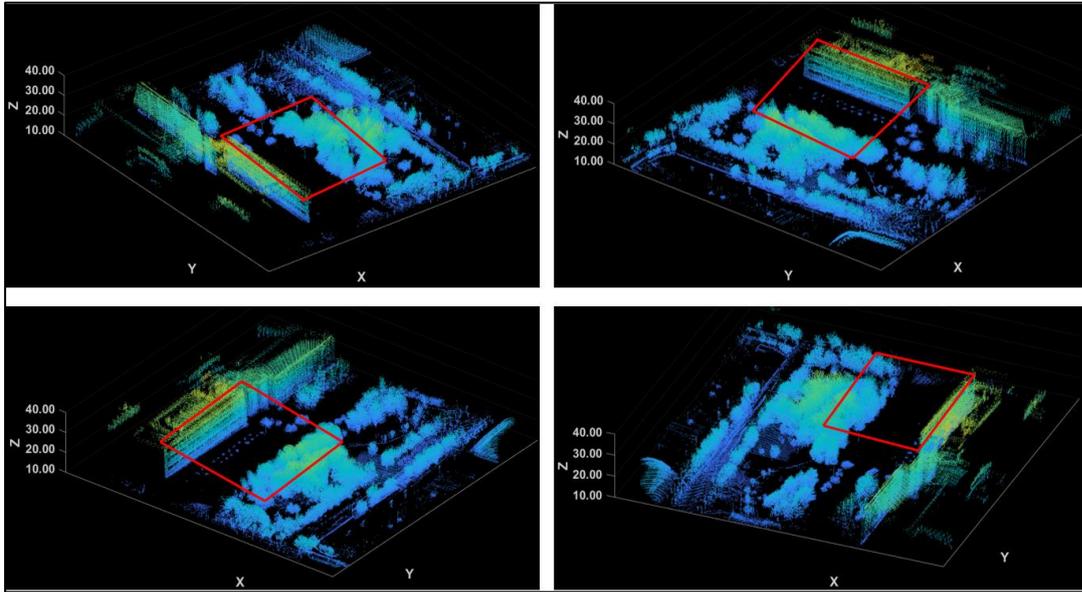


Figure 5.11 : Common point cloud map obtained with Jensen Divergence similarity metric output transform in the form of different camera line of sight. A part of these point cloud maps (red enclosed areas) can be compared with the given bird-eye view real image in Figure 5.4.

Also, the trajectory input and odometry output of the experiments are given in Figure 5.12. As it can be seen from the images and odometry outputs of ROS are similar. However, the ground robot could not realize the full path identical to the input trajectory due to the obstacles and traversability of the area.

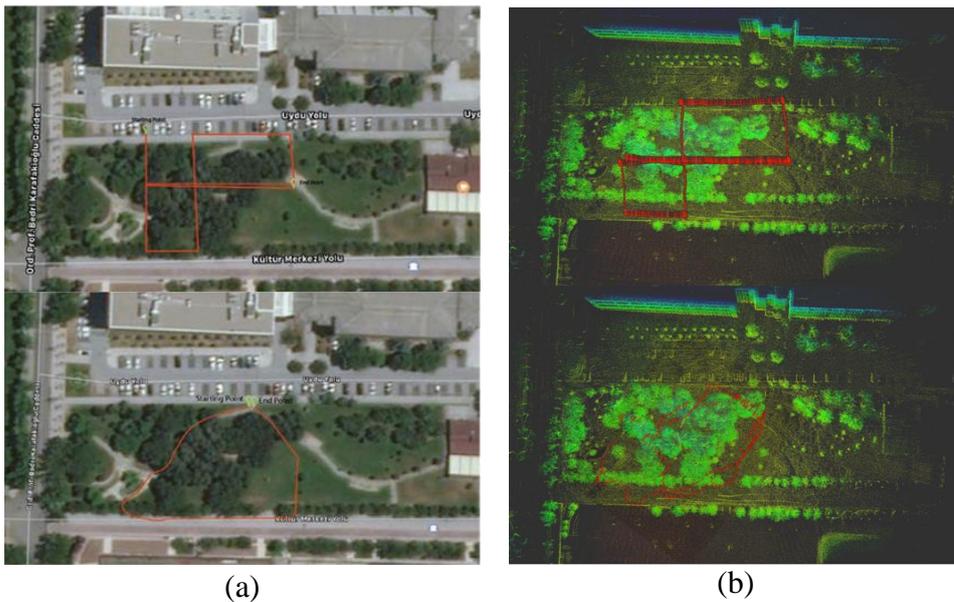


Figure 5.12 : Trajectory input and odometry output of the experiments.(a) taken from (Aybakan et al., 2019).

5.2 Simulation Studies

Simulation studies are used for validating the given theoretical background in this dissertation. In the following subsections, the usage of the simulation tools, scenario and results are explained and given in detail.

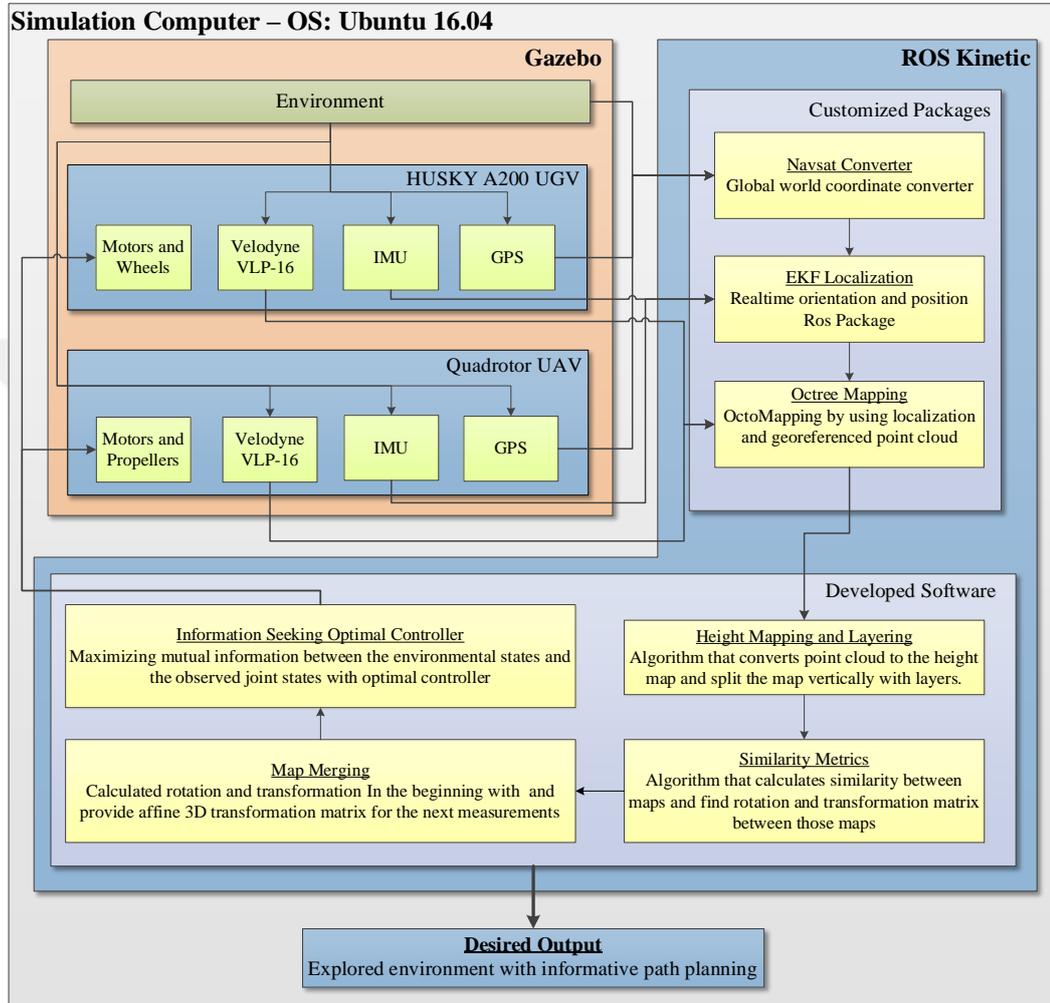


Figure 5.13 : Simulation setup functional block diagram.

5.2.1 Simulation Setup

The simulation setup includes tools that are utilized in this dissertation. The Robot Operating System (ROS) is employed to prevent time-wasting of reinventing the improvements that have already been achieved. It is an efficient and practical tool when developing robotic applications. ROS provides the flexibility of using specific algorithms with turnkey customizable infrastructure. Moreover, in order to design an environment and simulate the mission, Gazebo simulator is utilized, as well. This two software is available as open-source and can be used in Linux based operating systems,

such as; Ubuntu, Debian, etc. Finally, the algorithms of this dissertation are written on Python programming language, which is ROS compatible and easy to adapt with several useful tools. The functional block diagram of the simulations is given in Figure 5.13. The following subsections provide a comprehensive explanation of the usage of this software.

5.2.1.1 Robot Operating System

In this dissertation, ROS Kinetic is utilized for defining robots, collecting data, and processing them with Python algorithms and ready-to-use adjustable packages. UGV's wheels, body frame for move collision-free movement, sensor links and kinematic vehicle model are defined in ROS. Similar to that, UAV's specifications are implemented to ROS. However, the UAV model is not identical to the one used in experiments. Still, it has the same frame transformations and does the tasks of the mission identically. In Appendix A, obstacle avoidance (collision-free movement) method and kinematic vehicle models are given. EKF localization, OctoMap and Navsat Converter packages are used with custom parameters. The parameters, their values and the reasons for selected values are explained in Table 5.4.

Table 5.4 : ROS packages, parameters, their values and the reasons for selected values.

Package	Parameter	Value		Reason		
		UAV	UGV			
EKF Localization	frequency	40Hz	40Hz	Better to select 4 times faster than GPS data rate in order to correct measurements		
	frame	map_frame	map	map	-	
		odom_frame	odom	odom	Odom to base_link tf's are given.	
		base_link_frame	base_link	base_link	-	
		world_frame	odom	odom	-	
	sensor	gps	gps	gps	GPS, IMU and Husky A200 odometry data are collected in order to estimate the location of robots. UAV is modeled as constant-velocity, and wind disturbances are omitted.	
		imu	imu/data	imu/data		
		odometry	false	husky_velocity_controller / odom		
	sensor config	odomN_config	[0 0 0 0 0 0 0 0 0 0 0 0 0]	[1 1 0 0 0 0 0 0 0 0 0 0 0]	UGV encoder data are taken in accordance with the equation 2.3	
		twistN_config	[1 0 0 0 0 1 1 1 0 0 1 0 0]	[1 0 0 0 0 1 1 1 0 0 1 0 0]	Acceleration values are taken in order to create speed limits.	
		imuN_config	[0 0 0 0 0 0 0 0 1 1 1 0 0]	[0 0 0 0 0 0 0 0 1 1 1 0 0]	Angular velocities are taken in accordance with the equation 2.3	
		poseN_config	[1 1 1 1 1 1 0 0 0 0 0 0 0]	[1 1 1 1 1 1 0 0 0 0 0 0 0]	GPS sensors position values are taken in accordance with the equation 2.3	
	sensor_differential	imu, gps, encoder	false	false	No integration is needed. GPS data is converted through Navsat Converter.	
		use_control	true	true	Control commands are taken into account.	
		process_noise_covariance	$0.015 \times I_{9 \times 9}$	$0.015 \times I_{9 \times 9}$	Best results are obtained with these values.	
		dynamic_process_noise_covariance	false	false	Process noise is not scaled dynamically through velocities.	
		control_config	[1 1 1 0 0 1]	[1 0 0 0 0 1]	UAV uses three axis linear velocities and yaw, UGV is only uses linear velocity at heading angle and yaw value.	
	sensor noise	gps	drift	$0.01 \times I_{6 \times 6}$	$0.01 \times I_{6 \times 6}$	Best results are obtained with these values. Since sensors are virtual and not chosen related to the real sensors. Still, these values can be used in real-world studies.
			gaussian noise	$0.01 \times I_{6 \times 6}$	$0.01 \times I_{6 \times 6}$	
		imu	acc. drift	$0.05 \times I_{3 \times 3}$	$0.05 \times I_{3 \times 3}$	
acc.gaussian noise			$0.05 \times I_{3 \times 3}$	$0.05 \times I_{3 \times 3}$		

Table 5.4 (continued) : ROS packages, parameters, their values and the reasons for selected values.

Package	Parameter	Value		Reason	
		UAV	UGV		
OctoMap	rate drift	$0.05 \times I_{3 \times 3}$	$0.05 \times I_{3 \times 3}$	<p>The map is built on world frame</p> <p>Volumes occupied in accordance with base frame</p> <p>These resolutions are chosen related to the memory allocation</p> <p>UAV collects points on $\pm 135deg$ arc, UGV collects points on a 10m circle</p> <p>If the occupation probability reaches hit, it will be assigned as an occupied voxel, and vice versa, miss values will be assigned as a free voxel. 0.7 and 0.4 is an optimum value obtained by trials.</p> <p>These values are chosen related to the information loss.</p> <p>Same with EKF localization.</p> <p>Wait before calculating the GPS coordinates; it is related to sensor.</p> <p>It is calculated according to Turkey's latitude and longitude coordinates.</p> <p>IMU reads pi/2 when facing north. So, no need for offset for UGV. However, in real-time applications, UAV systems face IMU problems because of the full power of motors during elevation. So, an offset value is defined for UAV.</p>	
	rate				
	gaussian noise	$0.05 \times I_{3 \times 3}$	$0.05 \times I_{3 \times 3}$		
	heading drift	$0.05 \times I_{1 \times 1}$	$0.05 \times I_{1 \times 1}$		
	heading gaussian noise	$0.05 \times I_{1 \times 1}$	$0.05 \times I_{1 \times 1}$		
	frame_id	map_frame	world		world
		base_frame	base_link		base_link
		resolution	0.05-0.1-1		0.05-0.1-1
		sensor_model/max_range	25m		10m
		sensor_model/[hit miss]	0.7 / 0.4		
	sensor_model/[min max] frequency	0.1 / 0.96 – 0.25/0.7 – 0.4/0.6			
	delay	0.05sec	0.05sec		
Navsat Converter	magnetic_declination_radians	0.1rad	0.1rad		
	yaw_offset	0.26rad	0rad		

5.2.1.2 Gazebo Simulator

Gazebo Simulator 5.0 is used for modeling the environment similar to the real world by visually modeling of the simulation scene with already modeled objects and robots. The configuration of robots, sensors and physical attributes can be customized in order to emulate the real systems.

Simulation scene is designed to comprise the complexity of the environment with features, such as; trees, buildings, roofed top areas and traversable ground. The created simulation area can be seen in Figure 5.14.

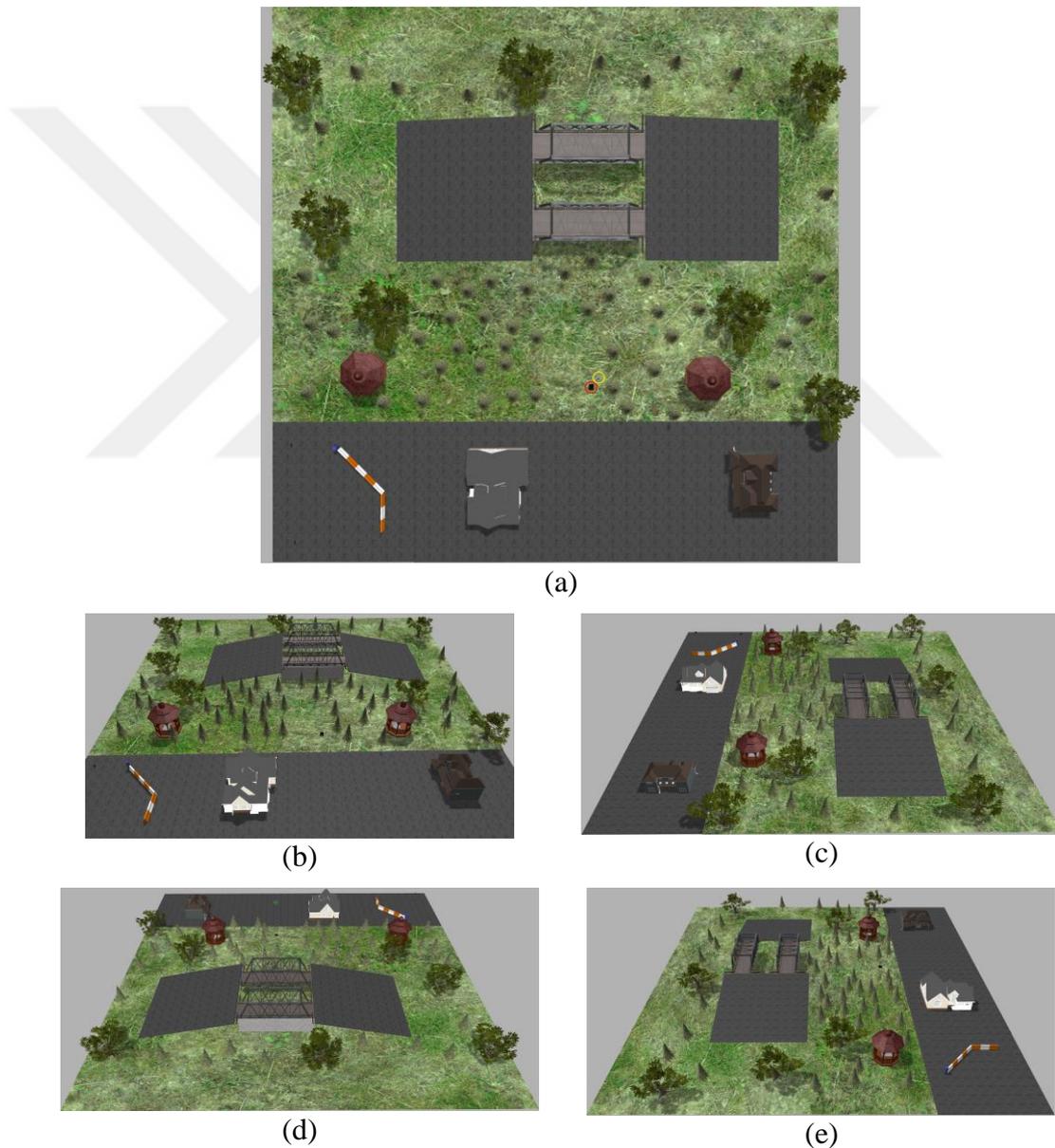


Figure 5.14 : Gazebo simulation scene for this dissertation. UGV and UAV are marked as red and yellow in (a), respectively. (a), top view. (b), (c), (d) and (e) views of the scene from different angles.

5.2.2 Scenario

Simulations are handled on Gazebo Simulator 5.0, ROS Kinetic and within the given scenario and conditions as following steps;

1. In the Gazebo scene, UGV and UAV locate side by side with $(x_d, y_d) = (0.3m, 1m)$ position difference as an initial condition.
2. Define the exploration area by giving the bounds of the area as GPS coordinates². Also, calculate the total volume of the explored space with, $V_e = 200m \times 200 \times 8$.
3. Send command: Elevate the UAV system to $z = 8m$ altitude.
4. Start Algorithm 2. Note that, this command is transmitted to obtain position and orientation difference of the local maps at the beginning of the simulation.

Algorithm 2 Calculating position and orientation difference

Input: $X_{k_{ugv}}(1:6)^{GPS}, X_{k_{uav}}(1:6)^{GPS}, X_{k_{ugv}}(7:9)^{IMU}, X_{k_{uav}}(7:9)^{IMU}, C_{(n,i)t_{uav}}^L, C_{(n,i)t_{ugv}}^L$

Output: \tilde{u}, H_j

begin Set initial target points on X-Y plane ($p_{1_{ugv}} = (0, 5, 0), p_{1_{uav}} = (0.3, 4, 8)$) for height map comparison.
 Georeference point cloud with EKF localization using $X_{k_{ugv}}(1:6)^{GPS}, X_{k_{uav}}(1:6)^{GPS}, X_{k_{ugv}}(7:9)^{IMU}, X_{k_{uav}}(7:9)^{IMU}, C_{(n,i)t_{uav}}^L, C_{(n,i)t_{ugv}}^L$
 Obtain $m_{t_{ugv}}$ and $m_{t_{uav}}$
 Calculate l_{UB_o} and l_{LB_o} with regard to desired output information
for $j = 1:4$ **do**
 for $i = 1:G$ **and do**
 Calculate $l(g_i|m_{1:t})_j$ for each layer j with
 if $P(g_i|m_t)_j > 0.3$;
 $z_{t_j} = z_{i_j}$
 else;
 $z_{t_j} = 0$
 end
 Obtain local the height map of UGV and UAV for each layer, $H_{0:t_{ugv}_j}$ and $H_{0:t_{uav}_j}$, respectively
 end
 #Initial value of u_i , with 10° rotation and $[x \ y]^T = [1m, 1m]^T$, translation difference
 $u_i = \begin{bmatrix} 0.98 & 0.17 & 1 \\ -0.17 & 0.98 & 1 \\ 0 & 0 & 1 \end{bmatrix}$
 for $i = i + 1$ **do**
 Calculate distances or entropies of the heights for each layer with the JeD.
 $\tilde{u}_j = \text{argmin } d_{JeD}$
 if \tilde{u}_j converges
 then; Break
 end
 Obtain \tilde{u}
 end
end

² GPS coordinates to global frame transform is given in Appendix C.

5. Start exploration of the environment with the Algorithm 3;

Algorithm 3 Active SLAM with Informative Path Planning for HeRT

Input: $P(w_s), I_{th}, X_{k_{ugv}}(1:6)^{GPS}, X_{k_{uav}}(1:6)^{GPS}, X_{k_{ugv}}(7:9)^{IMU}, X_{k_{uav}}(7:9)^{IMU}, C_{(n,i)t_{uav}}^L, C_{(n,i)t_{ugv}}^L$, Bounds of the area, \tilde{u} , Explored Space volume (V_e)

Output: $M_{w_c}, A_m, G_m, V_i, H_j, I(w_s, o_s)$

begin Elevate UAV for $8m$, set initial target points on X-Y plane ($p_{1_{ugv}}, p_{1_{uav}}$) for height map comparison
 Georeference point cloud with EKF localization using $X_{k_{ugv}}(1:6)^{GPS}, X_{k_{uav}}(1:6)^{GPS}, X_{k_{ugv}}(7:9)^{IMU}, X_{k_{uav}}(7:9)^{IMU}, C_{(n,i)t_{uav}}^L, C_{(n,i)t_{ugv}}^L$
 Obtain $m_{t_{ugv}}$ and $m_{t_{uav}}$
 Calculate l_{UB_o} and l_{LB_o} with regard to desired output information
 $S = V_e/V_s$
for $s = 1: S$ **and do**
 for $i = 1: V$ **do**
 Calculate $l(V_i|m_{1:t})$ with l_{UB_o} and l_{LB_o}
 Obtain local the OctoMap of UGV and UAV for each volumetric space, V_{ugv_s} and V_{uav_s} , respectively
 Obtain common OctoMap within the use of \tilde{u}
 end for
 for $t = 0: t_f$ **do**
 Calculate $P(o_s(t)), P(w_s(0: t_f), o_s(t))$
 Calculate $d_{KL}(P(w_s(0: t_f), o_s(t)) || P(w_s(0: t_f))P(o_s(t)))$
 if $d_{KL}(P(w_s, o_s) || P(w_s)P(o_s)) < I_{th}$;
 $O(u^r(t)) = \operatorname{argmax}(d_{KL}(P(w_s(0: t_f), o_s(t)) || P(w_s(0: t_f))P(o_s(t))))$
 $u_t^r = \omega_s \frac{\partial O_t}{\partial p_{c_t}}$
 Obtain p_{c_i}
 if $p_{c_i}^z \leq h_2$;
 $p_{c_t} = p_{c_{ugv_t}}$
 else;
 $p_{c_t} = p_{c_{uav_t}}$
 end
 for $k = 0: 4$ **do** #reachability check
 if $p_{c_{ugv_{t-1}}} \neq X_{k_{ugv}}(1:2)_t \ \&\& \ X_{k_{ugv}}(1:2)_t - X_{k_{ugv}}(1:2)_{t-1} \geq D_{tolerance}$;
 $p_{c_{ugv_{t-1}}} = p_{c_{uav_{t+1}}}$
 end
 end
 Move to target point within creating collision-free path
 else;
 Break
 end for
end for
end
 Height map of the environment within the use of OctoMap center points
end

5.2.3 Results

Results are obtained with different cases in order to realize the advantages of this dissertation. The information about the cases is given in Table 5.5 rest of the parameters are as in Table 5.4.

To keep simplicity and compatibility, height map bounds, distance tolerance and volume of the investigation spaces are all the same for six cases. The threshold value

of mutual information between environmental states and observed states, I_{th} for each space is defined by $P(w_s)$ which takes its values from $P(w_s) > p_1$ or $P(w_s) < p_2$.

Table 5.5 : Parameters were used in cases of the study.

Case No.	1	2	3	4	5	6
Robots	UAV	UGV	HeRT	HeRT	HeRT	HeRT
l_{LB_h}	-2	-2	-2	-2	-2	-2
l_{UB_h}	3	3	3	3	3	3
$D_{tolerance}$ (m)	-	-	2	2	2	2
l_{LB_o}	-2.2	-2.2	-2.2	-1.1	-0.8	-2.2
l_{UB_o}	3.1	3.1	3.1	0.8	0.6	3.1
OctoMap Resolution (m)	0.05	0.05	1	0.1	0.05	0.05
$a \times a \times b = V_s$ (m)	$5 \times 5 \times 2$	$5 \times 5 \times 2$	$5 \times 5 \times 2$	$5 \times 5 \times 2$	$5 \times 5 \times 2$	$5 \times 5 \times 2$
I_{th} (bits)	0.3	0.3	0.24	0.24	0.33	0.33
p_1	0.9	0.9	0.8	0.8	0.93	0.93
p_2	0.1	0.1	0.2	0.2	0.07	0.07

In simulation studies, only JeD is used for calculating the position and orientation difference, since it is defined as the best similarity metric in experimental studies. For cases 3-6, the affine 3D transformation matrix is calculated as 15° rotation and $(x_d, y_d) = (0.31m, 0.98m)$ position difference, and it is used for map merging during the whole process. Very close values between the results and given simulation inputs are achieved.

The rest of the results are explained by visual and by exploration performances as follows.

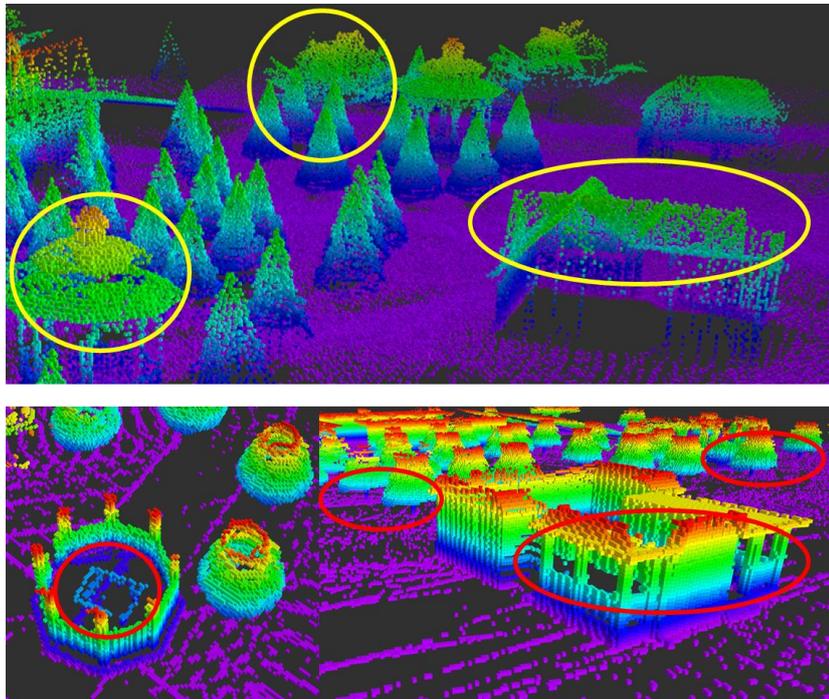


Figure 5.15 : Advantages of employing HeRT. Yellow areas show observations of UAV in case 1. Red areas show observations of UGV in case 2.

5.2.3.1 Visual Outputs

The quality of the obtained maps and the advantages of the used method can be explained by using visual outputs.

Employing HeRT instead of only UGV or UAV provides a lot of details that can be seen in Figure 5.15. It is possible to observe under the roofed top areas such as gazebos and houses (inside the windows). Also, information about the upper side of the trees and roofs is perceived.

Moreover, the total height map of the environment can be seen in Figure 5.16.

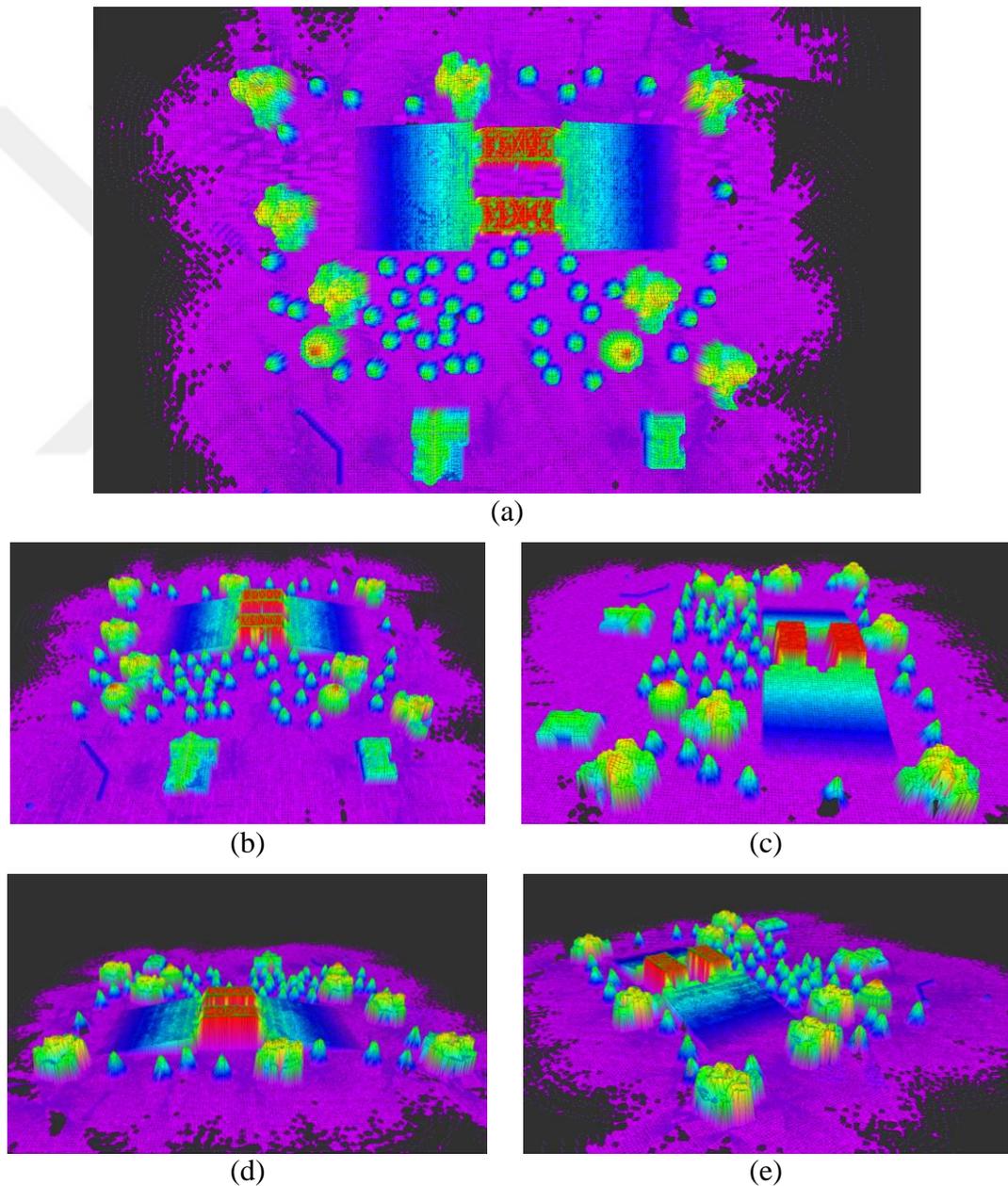


Figure 5.16 : Obtained height map of the environment by HeRT. (a) to (f), with different view angles.

Obtained OctoMaps of the case studies are given in Figure 5.17. So, in case 6, the best visually compatible result is achieved with higher information threshold and wide boundary values. Also, lower resolution values of the voxels in Octomap supply a more detailed map. However, these values lead to the necessity of more memory because of the obtained details.

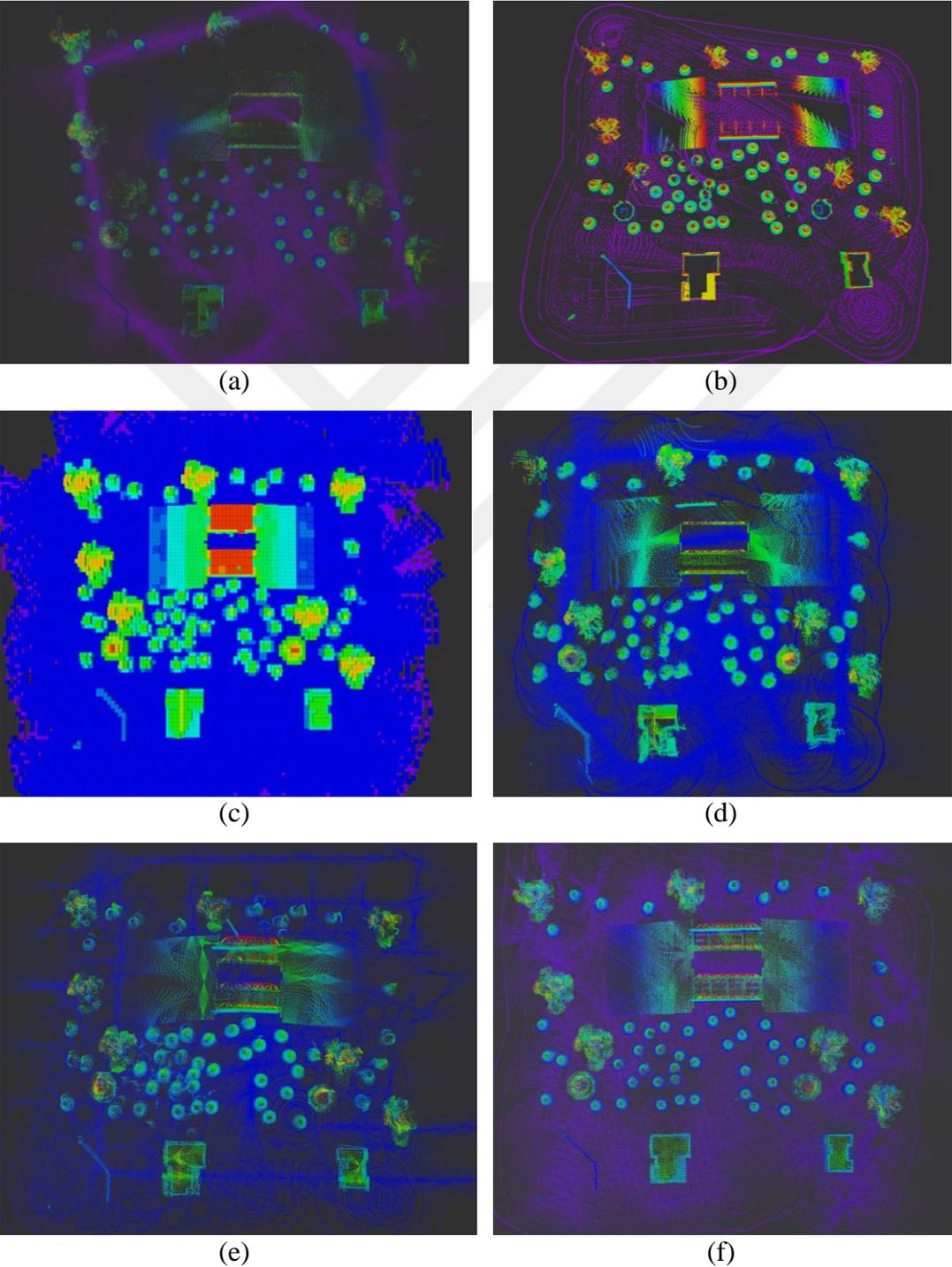


Figure 5.17 : Obtained OctoMaps of the environment from (a) to (f), with regard to cases 1 to 6, respectively.

Additionally, some of the details cannot be observed, and distortions are realized with the use of lower information threshold in case 4 and strict OctoMap permanent occupancy lower and upper bounds in cases 4 and 5 compared to case 6, as shown in Figure 5.18.

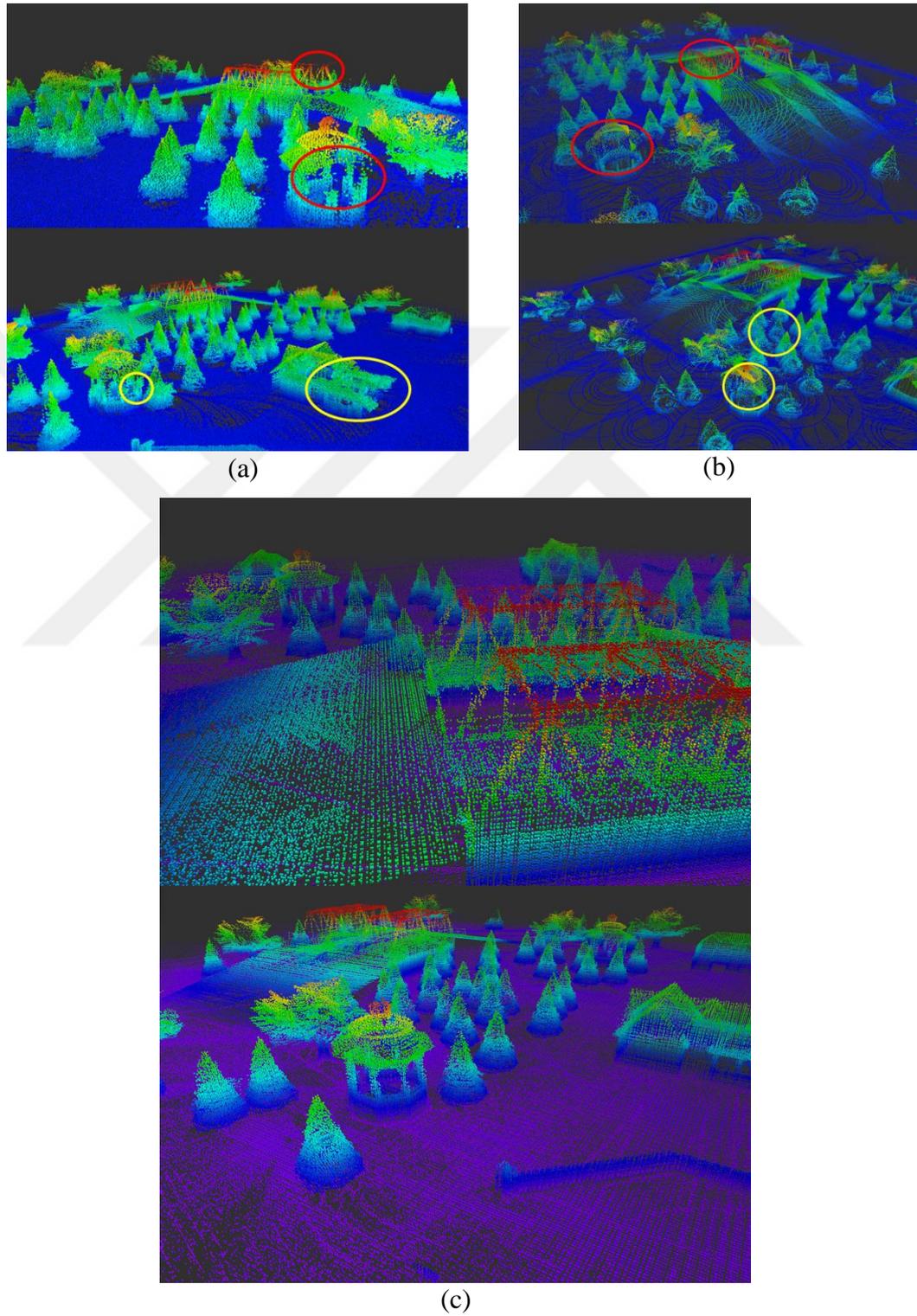


Figure 5.18 : Lack of information (red circles) and distortions of visual outputs (yellow circles). (a) case 4. (b) case 5. (c) case 6.

5.2.3.2 Performance Outputs

The exploration performances of cases are summed up in Table 5.6. Note that, speeds of the UGV and UAV are limited to 0.5m/s and 1m/s, respectively. Also, UAV will turn around itself at the $p_{c_{uav_t}}$ for maximum two times in order to attain maximum information on the volumetric space until the information value is above the threshold.

Table 5.6 : Exploration performances of cases.

Case No.	Duration (mins)	Memory usage (MB)	Collected total information (bits)	VSLUT/VSE (%)	Total Movement (m)	
					UGV	UAV
1	59	156	1140	%38	-	2832
2	105	115	845	%54	3005	-
3	69	0.224	1250	%26	1570	1032
4	81	32	1377	%22	1440	1825
5	117	228	1774	%20	1755	1930
6	87	237	2107	%17	1875	2035

For the purpose of investigating the total movement and volumetric spaces left under the threshold (VSLUT) according to the total number of volumetric spaces of the environment (VSE), the extracted performance metrics are given in Figure 5.19 as a chart.

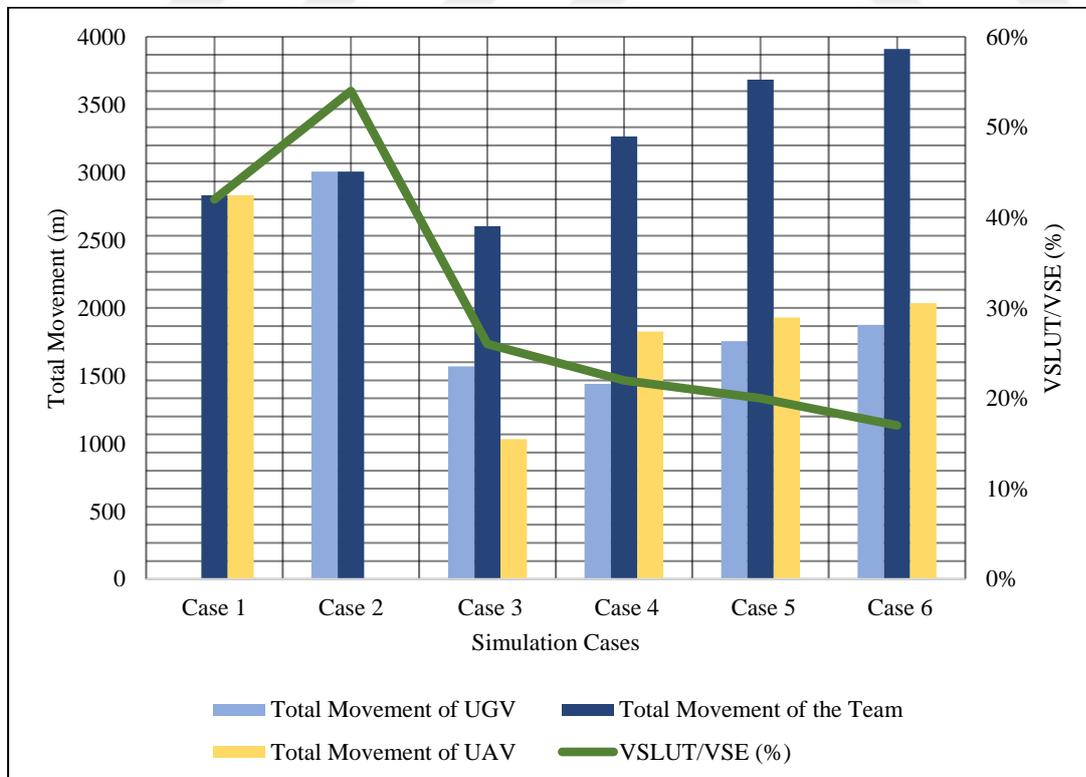


Figure 5.19 : Total movement of the team and volumetric spaces left under the threshold according to the total number of volumetric spaces, case by case.

As can be seen from Figure 5.19, employing only UAV provides less effort in terms of total movement with better area coverage. On the other hand, in the cases that HeRT is utilized (case 4 to 6), the UAV movement is higher than the UGV's. The reason why this happened is UAV observes areas with its measurements faster than the UGV and UAV receives more information maximization control input. So, UAV is moving to more points than the UGV in order to cover the area and exceed the threshold on the volumetric spaces of the environment. Case 3 is an exception because the UAV system can measure distances in a wider area. There is not any necessity of moving to the target points, since it is observed from farther points with the less resolution of OctoMap. Further, the uncovered volumetric spaces decrease while the total movement of the team increases. This performance evolution is compatible with the expected results considering covering more areas.

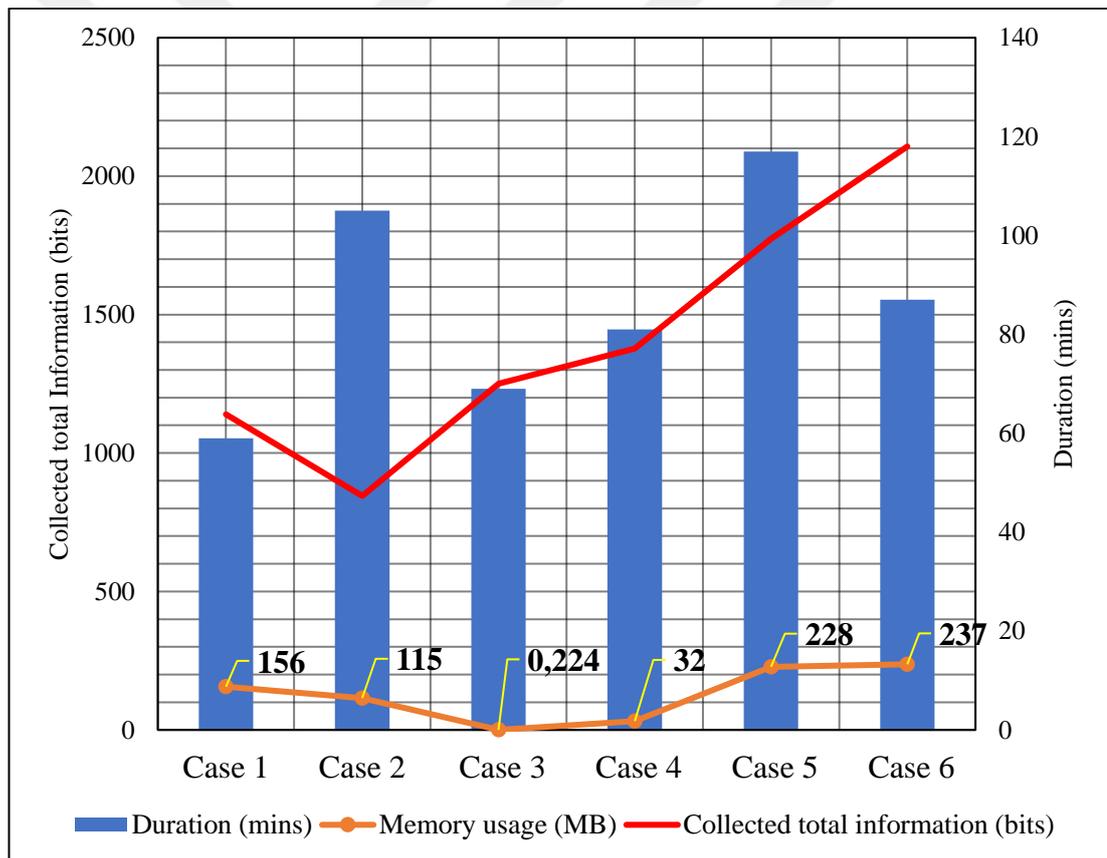


Figure 5.20 : Evolution of collected total information, duration of the mission and the memory usage of the obtained map, case by case.

Figure 5.20 shows that the resolution of OctoMap directly affects the memory usage since the shared map is an OctoMap file with “.ot” extension. In addition, collected information is also increasing the memory allocation, but its effect is much smaller than the resolution. Moreover, to explain the results in terms of duration, case 1 and

case 2 can be compared. The duration of case 1 is less than case 2, since UAV can move faster than the UGV and do not need any collision-free path planning. Also, the duration of case 5 is higher than all of the cases. It is because the lower and upper bound values of OctoMap are so strict as consequence robots spent more time bringing the collected information value over its threshold. This is one of the optimized features of the framework presented in this dissertation. With the greater bounds of permanent occupancy probability, the total amount of collected information can be higher with less duration than case 5, as in case 6. Another outcome of the more limited bounds is encountering more VSLUT over the environment. Although the information threshold is the same in case 5 and case 6, the collected total information about the environment is less due to the limited bounds.

In case 4 and case 3, the duration is better than the duration of case 5 and case 6, and the collected total information is acceptable; however, more VSLUT over the environment is achieved. Nevertheless, the collected total information in case 3 is not practicable because of the OctoMap resolution. With this value of resolution, the traversability check, obstacle avoidance, etc. are extremely difficult. As proof of this, the UGV movement is higher than in case 4 because of trying to reach the target point several times, even though the bounds are greater in case 3. Besides, VSLUT is greater than in cases 4, 5 and 6. In case 6, the best result is obtained in terms of collected information with the optimal duration of the mission and lower VSLUT. Only the memory usage may appear high in case 6 for readers, but it provides more information and a utilizable map for traversability check and obstacle avoidance. Further, the only reason for higher movement in case 6 is, robots' attempt to reach more points as it can be seen from achieving the least VSLUT.

Moreover, employing the HeRT provides higher collected total information with less duration of the mission compared to case 2. The duration is the least in case 1 by sacrificing volumetric spaces achieved over the threshold with respect to cases 3 to 6. UAV usage, as in case 1, leads to obtain less amount of information because of UAV's sight of view.

The entropy of the whole map during simulation is decreasing vice versa the mutual information between map and observations is increasing. The evolution of the mutual information between the environmental states and observations during the exploration mission are given case by case in Figure 5.21. This figure shows that the information

value is logarithmically increasing since all the thresholds on the volumetric spaces are exceeded. The evolution of the information is not perfectly continuous due to the times spend for robots movement to specific target points and waiting to collect adequate information on that point. On the contrary, at some target points, the robot may not be able to observe areas behind the objects. As a result of that, with passing around the object, sudden observations provide a rapid increase of the obtained information from the area behind the object.

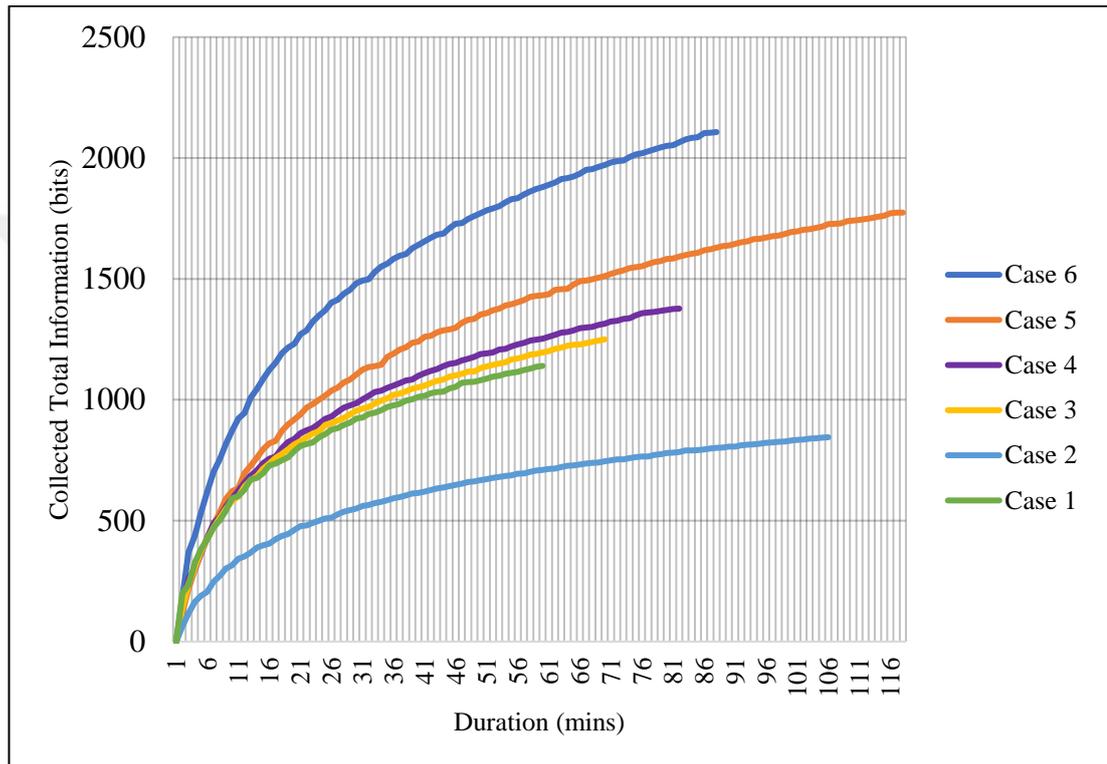


Figure 5.21 : The entropy evolution during the exploration mission, case by case.

Outputs of the results can be listed as follows;

1. l_{LB_0} and l_{UB_0} values affect the duration of the exploration mission and collected information,
2. OctoMap resolution mostly acts as the memory usage reducing parameter of the framework. However, it increases the total movement of the robots, which is the energy consumption of the systems.
3. I_{th} values determine the collected total information over the environment. Also, the threshold has an impact on the total movement of the robots,
4. Employing HeRT instead of one robot provides better exploration performance in accordance with collected total information and duration of the mission.

In conclusion, the theoretic approaches with the given framework are validated through the real-time experiments and demonstrations in the simulation environment. HeRT successfully explored the given environment with the use of the adjustable framework given in this dissertation. The next section serves to summarize the dissertation and discuss the achievements of the presented framework.



6. CONCLUSION

Understanding the environment in the same manner for UGV and UAV robots in HeRT missions can be difficult because of robots' various specifications such as moveable paths, the sight of view and onboard sensors. Common height maps with various entropy-based similarity metrics are compared by this dissertation to understand the environment with complex texture for both ground and aerial vehicles, even in lack of GPS sensors, thanks to the LOAM method. This thesis provides a method for exploring the environment in cases where GPS signals are jammed or blocked. As a result of this feature, the method can also be used indoors, caves, GPS jammed risky territories, etc.

Calculating rotation and translation between two height maps obtained by robots with different sights of view brings better awareness of the environment rather than using only point clouds or 2D grid maps. Also, if it is needed to go through a roofed area or under a tree by the aerial robot, this method will provide practical information with fewer errors. On the other hand, employing only the height map approach may not be enough to plan trajectories in the environment or to achieve a better awareness for all applications. The height values of the grids can be obtained differently from each HeRT members because of their sight of view. Unlike the rest of the studies in literature, the layering method is utilized in the experiments of this dissertation, and this method significantly affected this study by constructing a %44 better map for the best similarity metric (JeD) in terms of RMSE with average %15 faster computation time. Due to the complex texture of the experimental area given in this dissertation, it is understood that the layering method is needed for obtaining a more accurate common map.

To sum up the constructing common map framework, with the experiments, features located beneath the trees or the roofed top areas and above them are observed without any need for GPS signal. Additionally, a more effective common map that enables planning trajectories for both vehicles is obtained with the determined similarity metric and the layering method.

Path planning for HeRT is another challenging topic in robotic navigation in terms of providing optimal control commands to the robots. With this study, a novel approach to informative path planning is presented and validated with the simulations.

Defining information thresholds for path planning ensures adjustable outputs such as; exploration time, memory allocation and visual details. These outputs, which are likely to change from application to application, are one of the prominent contributions of this study. Information based path planning control inputs may force the robots to move the designated points in any case without checking the reachability. Volumetric space allocation by robots offers a simple and easily implementable framework with specified rules in contrast to consensus-based algorithms. Also, conflicts on the decision making of UGV and UAV are omitted with these rules. In addition, volumetric spaces provide scalable computation complexity by changing the dimensions of the volume. In this study, within tuning the parameters average 53% better information obtained about the environment with the cost of 5.7% longer duration for exploration (case 6) compared to employing only one robot (case 1 and case 2). On the other hand, 33% shorter duration and 28% less action for robots with a 46% decrease of collected information on exploration mission are achieved with the comparison between employing only UAV (case 2) and HeRT (case 6).

In conclusion, exploring a defined area with complex features by employing informative path planning method for HeRT is validated with the simulations. Utilizing HeRT, layering method, various mapping technique and similarity metrics for common map constructing, relative entropy theory, and obtaining a Lyapunov stable optimal controller with specific rules provides adaptive and effective Active SLAM framework for the exploration of the environment.

6.1 Practical Applications of This Study

This study can be used for autonomously exploring the risky territories, forests, caves, etc. with tuning the parameters in order to carry out the mission requirements. The risky territories such as; desolated areas, dysfunctional nuclear power plants and wreckages are dangerous for humans to inspect; hence sacrificeable robots can be employed.

This infrastructure is used to lower the risk with only possible tangible loss. For example, after the earthquake, an area where the bounds are known lined with wreckage. It is very risky to search the area with humans, since building parts may still fall and harm people. However, HeRT can be employed in order to check the health of the construction with visual inspection without any human interaction.

Further, this study provides a great tool for investigating hostile territories such as caves, forests, or a plant where GPS signals may jammed or blocked. To exemplify, in order to move the troops to a specific location, the path and around the trajectory must be known beforehand. This framework provides exploration of the area to control the traversability of the path without any need of a human in-the-loop system.

6.2 Future Work

More than two robots, such as; three UGV and four UAV or swarms, can be employed to explore the area with less time. This improvement causes scalability problems and brings necessity of consensus-based algorithms. Within the implementation of these algorithms and overcoming the scalability problem, this framework will be a great utility for more effective exploration missions.

In addition, to unmount the ground station from the working principle of this framework, decentralized or distributed infrastructure can be applied. Nevertheless, communication constraints and optimal communication-based algorithms must be defined as in (Imer et al., 2006). As a result, the system will not be affected by communication blockage.



REFERENCES

- Akay, M. C., Aybakan, A., & Temeltas, H.** (2018). Map Matching Performance under Various Similarity Metrics for Heterogeneous Robot Teams. *International Journal of Electrical, Electronic and Communication Sciences*, 12(12), 884–888. <https://doi.org/10.5281/zenodo.2022110s>
- Akay, M. C., & Temeltas, H.** (2020). Constructing common height maps with various entropy-based similarity metrics and utilizing layering method for heterogeneous robot teams. *Industrial Robot: The International Journal of Robotics Research and Application*, 47(6), 889-902. <https://doi.org/10.1108/IR-03-2020-0062>
- Atanasov, N., Le Ny, J., Daniilidis, K., & Pappas, G. J.** (2015). Decentralized active information acquisition: Theory and application to multi-robot SLAM. *Proceedings - IEEE International Conference on Robotics and Automation, 2015-June*(June), 4775–4782. <https://doi.org/10.1109/ICRA.2015.7139863>
- Aybakan, A., Haddeler, G., Akay, M. C., M., Ervan, O., & Temeltas, H.** (2019). A 3D LiDAR dataset of ITU heterogeneous robot team. *ACM International Conference Proceeding Series*, 12–17. <https://doi.org/10.1145/3373724.3373734>
- Binney, J., & Sukhatme, G. S.** (2012). Branch and bound for informative path planning. *Proceedings - IEEE International Conference on Robotics and Automation*, 2147–2154. <https://doi.org/10.1109/ICRA.2012.6224902>
- Bosse, M., & Zlot, R.** (2009). Continuous 3D scan-matching with a spinning 2D laser. *2009 IEEE International Conference on Robotics and Automation*, 4312–4319. <https://doi.org/10.1109/robot.2009.5152851>
- Bosse, M., Zlot, R., & Flick, P.** (2012). Zebedee: Design of a spring-mounted 3-D range sensor with application to mobile mapping. *IEEE Transactions on Robotics*, 28(5), 1104–1119. <https://doi.org/10.1109/TRO.2012.2200990>
- Bryson, M., & Sukkarieh, S.** (2008). Observability analysis and active control for airborne SLAM. *IEEE Transactions on Aerospace and Electronic Systems*, 44(1), 261–280. <https://doi.org/10.1109/TAES.2008.4517003>
- Capitán, J., Merino, L., Caballero, F., & Ollero, A.** (2011). Decentralized Delayed-State Information Filter (DDSIF): A new approach for cooperative decentralized tracking. *Robotics and Autonomous Systems*, 59(6), 376–388. <https://doi.org/10.1016/j.robot.2011.02.001>
- Carlone, L., Du, J., Kaouk Ng, M., Bona, B., & Indri, M.** (2010). An application of Kullback-Leibler divergence to active SLAM and exploration with particle filters. *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, 287–293. <https://doi.org/10.1109/IROS.2010.5652164>

- Carlone, L., Du, J., Kaouk Ng, M., Bona, B., & Indri, M.** (2014). Active SLAM and exploration with particle filters using Kullback-Leibler divergence. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 75(2), 291–311. <https://doi.org/10.1007/s10846-013-9981-9>
- Cha, S.** (2007). *Comprehensive Survey on Distance / Similarity Measures between Probability Density Functions*. 1(4).
- Durrant-Whyte, H., & Bailey, T.** (2006). Simultaneous localization and mapping: Part I. *IEEE Robotics and Automation Magazine*, 13(2), 99–108. <https://doi.org/10.1109/MRA.2006.1638022>
- Fairfield, N., Kantor, G., & Wettergreen, D.** (2007). Real-time SLAM with octree evidence grids for exploration in underwater tunnels. *Journal of Field Robotics*, 24(1–2), 03–21. <https://doi.org/10.1002/rob.20165>
- Forster, C., Pizzoli, M., & Scaramuzza, D.** (2013). Air-ground localization and map augmentation using monocular dense reconstruction. *IEEE International Conference on Intelligent Robots and Systems*, 3971–3978. <https://doi.org/10.1109/IROS.2013.6696924>
- Fox, D., Burgard, W., & Thrun, S.** (1997). The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, 4(1), 23–33. <https://doi.org/10.1109/100.580977>
- Fu, C., Mertz, C., & Dolan, J. M.** (2019). LIDAR and Monocular Camera Fusion: On-road Depth Completion for Autonomous Driving. *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019*, 273–278. <https://doi.org/10.1109/ITSC.2019.8917201>
- Gorji, A., Menhaj, M. B., & Shiry, S.** (2007). Multiple target tracking for mobile robots using the JPDAF algorithm. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI, 1*, 137–145. <https://doi.org/10.1109/ICTAI.2007.11>
- Haddeler, G., Aybakan, A., Akay, M. C., & Temeltas, H.** (2020). Evaluation of 3D LiDAR Sensor Setup for Heterogeneous Robot Team. *Journal of Intelligent & Robotic Systems*. <https://doi.org/10.1007/s10846-020-01207-y>
- Hofmeister, M., Kronfeld, M., & Zell, A.** (2011). Cooperative visual mapping in a heterogeneous team of mobile robots. *Proceedings - IEEE International Conference on Robotics and Automation*, 1491–1496. <https://doi.org/10.1109/ICRA.2011.5979558>
- Hollinger, G. a., & Sukhatme, G. S.** (2013). Stochastic motion planning for robotic information gathering. *Robotics: Science and Systems (RSS), Section V*. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.309.7308&rep=rep1&type=pdf>
- Hood, S., Benson, K., Hamod, P., Madison, D., O’Kane, J. M., & Rekleitis, I.** (2017). Bird’s eye view: Cooperative exploration by UGV and UAV. *2017 International Conference on Unmanned Aircraft Systems, ICUAS 2017, September*, 247–255. <https://doi.org/10.1109/ICUAS.2017.7991513>

- Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W.** (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3), 189–206. <https://doi.org/10.1007/s10514-012-9321-0>
- Howard, A.** (2006). Multi-robot simultaneous localization and mapping using particle filters. *International Journal of Robotics Research*, 25(12), 1243–1256. <https://doi.org/10.1177/0278364906072250>
- Huang, S., Kwok, N. M., Dissanayake, G., Ha, Q. P., & Fang, G.** (2005). Multi-step look-ahead trajectory planning in SLAM: Possibility and necessity. *Proceedings - IEEE International Conference on Robotics and Automation, 2005(May)*, 1091–1096. <https://doi.org/10.1109/ROBOT.2005.1570261>
- Husain, A., Jones, H., Kannan, B., Wong, U., Pimentel, T., Tang, S., Daftry, S., Huber, S., & Whittaker, W. L.** (2013). Mapping planetary caves with an autonomous, heterogeneous robot team. *IEEE Aerospace Conference Proceedings*. <https://doi.org/10.1109/AERO.2013.6497363>
- Husky UGV - Outdoor Field Research Robot by Clearpath.** (n.d.). Retrieved July 7, 2020, from <https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/#tech-specs>
- Imer, O. C., Yuksel, S., & Basar, T.** (2006). Optimal control of LTI systems over unreliable communication links. *Automatica*, 42(9), 1429–1439. <https://doi.org/10.1016/j.automatica.2006.03.011>
- Indelman, V.** (2015). Towards multi-robot active collaborative state estimation via belief space planning. *IEEE International Conference on Intelligent Robots and Systems, 2015-Decem*, 4620–4626. <https://doi.org/10.1109/IROS.2015.7354035>
- Indelman, V., Carlone, L., & Dellaert, F.** (2014). Planning under uncertainty in the continuous domain: A generalized belief space approach. *Proceedings - IEEE International Conference on Robotics and Automation, June*, 6763–6770. <https://doi.org/10.1109/ICRA.2014.6907858>
- Indelman, V., Carlone, L., & Dellaert, F.** (2015). Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments. *The International Journal of Robotics Research*, 34(7), 849–882. <https://doi.org/10.1177/0278364914561102>
- Jose Luis Blanco Claraco, J.-A. F.-M.** (2012). *Simultaneous Localization and Mapping for Mobile Robots: Introduction and Methods*.
- Julian, B. J., Angermann, M., Schwager, M., & Rus, D.** (2012). Distributed robotic sensor networks: An information-theoretic approach. *The International Journal of Robotics Research*, 31(10), 1134–1154. <https://doi.org/10.1177/0278364912452675>
- Kaeslin, R., Stumm, E., Taylor, Z., Elias, M., Delmerico, J., Scaramuzza, D., Siegwart, R., & Hutter, M.** (2016). Collaborative Localization of Aerial and Ground Robots through Elevation Maps. *IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. <https://doi.org/10.3929/ethz-a-010723041>

- Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., & Dellaert, F.** (2012). iSAM2: Incremental smoothing and mapping using the Bayes tree(graph optimization, bayes). *International Journal of Robotics Research*, 31(2), 216–235. <https://doi.org/10.1177/0278364911430419>
- Kleiner, A., & Dornhege, C.** (2007). Real-time localization and elevation mapping within urban search and rescue scenarios. *Journal of Field Robotics*, 24(8–9), 723–745. <https://doi.org/10.1002/rob.20208>
- Kneip, L., Chli, M., & Siegwart, R. Y.** (2011). Robust Real-Time Visual Odometry with a Single Camera and an IMU. *Bmvc*, 16.1-16.11. <https://doi.org/10.5244/C.25.16>
- Kontitsis, M., Theodorou, E. A., & Todorov, E.** (2013). Multi-robot active SLAM with relative entropy optimization. *Proceedings of the American Control Conference*, 2757–2764. <https://doi.org/10.1109/acc.2013.6580252>
- Ktiri, Y., & Inaba, M.** (2012). A framework for multiple heterogeneous robots exploration using laser data and MARG sensor. *2012 IEEE/SICE International Symposium on System Integration, SII 2012*, 635–640. <https://doi.org/10.1109/SII.2012.6426946>
- Kullback, S., & Leibler, R. A.** (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 79–86.
- Kumar, V., Rus, D., & Singh, S.** (2004). Robot and Sensor Networks for First Responders. *IEEE Pervasive Computing*, 3(4), 24–33. <https://doi.org/10.1109/MPRV.2004.17>
- Langerwisch, M., Wittmann, T., Thamke, S., Remmersmann, T., Tiderko, A., & Wagner, B.** (2013). Heterogeneous teams of unmanned ground and aerial robots for reconnaissance and surveillance-a field experiment. *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium On*, 1–6.
- Łazaro, M. T., Paz, L. M., Pinies, P., & Castellanos, J. A.** (2013). Distributed localization and submapping for robot formations using a prior map. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 8(PART 1), 138–145. <https://doi.org/10.3182/20130626-3-AU-2035.00022>
- Li, Y., & Ruichek, Y.** (2014). Occupancy grid mapping in urban environments from a moving on-board stereo-vision system. *Sensors (Switzerland)*, 14(6), 10454–10478. <https://doi.org/10.3390/s140610454>
- Lourenco, P., Batista, P., Oliveira, P., & Silvestre, C.** (2015). Towards uncertainty optimization in active SLAM. *Proceedings of the IEEE Conference on Decision and Control, 54rd IEEE*, 3242–3247. <https://doi.org/10.1109/CDC.2015.7402706>
- Makarenko, A. A., Williams, S. B., Grocholsky, B., & Durrant-whyte, H. F.** (2006). Information Based Adaptive Robotic Exploration. *IEEE/RSJ International Conference on Intelligent Robots and Systems, October 2002*, 540–545.
- Mao, X., Kunii, T. L., Fujishiro, I., & Noma, T.** (1987). Hierarchical Representations of 2D/3D Gray-Scale Images and Their 2D/3D Two-Way Conversion. *IEEE Computer Graphics and Applications*, 7(12), 37–44. <https://doi.org/10.1109/MCG.1987.276937>

- Martinez-Cantin, R., De Freitas, N., Brochu, E., Castellanos, J., & Doucet, A.** (2009). A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Autonomous Robots*, 27(2), 93–103. <https://doi.org/10.1007/s10514-009-9130-2>
- Martinez-cantin, R., Freitas, N. De, & Doucet, A.** (2008). Active Policy Learning for Robot Planning and Exploration under Uncertainty. In *Robotics* (pp. 321–328). The MIT Press. <https://doi.org/10.7551/mitpress/7830.003.0041>
- Matrice 600 Pro Specs, FAQ, Tutorials and Downloads - DJI. (n.d.)**. Retrieved July 7, 2020, from <https://www.dji.com/matrice600-pro/info#specs>
- Meyer, F., Wymeersch, H., Frohle, M., & Hlawatsch, F.** (2015). Distributed estimation with information-seeking control in agent networks. *IEEE Journal on Selected Areas in Communications*, 33(11), 2439–2456. <https://doi.org/10.1109/JSAC.2015.2430519>
- Nam, T. H., Shim, J. H., & Cho, Y. I.** (2017). A 2.5D map-based mobile robot localization via cooperation of aerial and ground robots. *Sensors (Switzerland)*, 17(12), 1–24. <https://doi.org/10.3390/s17122730>
- Nurmaini, S., & Tutuko, B.** (2017). Intelligent robotics navigation system: Problems, methods, and algorithm. *International Journal of Electrical and Computer Engineering*, 7(6), 3711–3726. <https://doi.org/10.11591/ijece.v7i6.pp3711-3726>
- Palomar, D. P., & Verdú, S.** (2007). Representation of mutual information via input estimates. *IEEE Transactions on Information Theory*, 53(2), 453–470. <https://doi.org/10.1109/TIT.2006.889728>
- Parker, L. E., Kannan, B., Tang, F. T. F., & Bailey, M.** (2004). Tightly-coupled navigation assistance in heterogeneous multi-robot teams. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, 1, 1016–1022. <https://doi.org/10.1109/IROS.2004.1389486>
- Pham, V. C., & Juang, J. C.** (2013). A multi-robot, cooperative, and active SLAM algorithm for exploration. *International Journal of Innovative Computing, Information and Control*, 9(6), 2567–2583.
- Sanfeliu, A., & Andrade-Cetto, J.** (2006). Ubiquitous Networking Robotics in Urban Settings. *Proceedings of the IEEE/RSJ IROS Workshop on Network Robot Systems*, 14–18. <http://hdl.handle.net/10261/30319>
- Scherer, S., Rehder, J., Achar, S., Cover, H., Chambers, A., Nuske, S., & Singh, S.** (2012). River mapping from a flying robot: State estimation, river detection, and obstacle mapping. *Autonomous Robots*, 33(1–2), 189–214. <https://doi.org/10.1007/s10514-012-9293-0>
- Segal, A., Haehnel, D., & Thrun, S.** (2009, June 28). Generalized-ICP. *Robotics: Science and Systems V*. <https://doi.org/10.15607/RSS.2009.V.021>
- Shannon, C. E.** (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(4), 623–656. <https://doi.org/10.1002/j.1538-7305.1948.tb00917.x>
- Sileshi, B. G., Ferrer, C., & Oliver, J.** (2013). Particle filters and resampling techniques: Importance in computational complexity analysis. *Conference on Design and Architectures for Signal and Image Processing, DASIP, 1*, 319–325.

- Sim, R., & Roy, N.** (2005). Global a-optimal robot exploration in SLAM. *Proceedings - IEEE International Conference on Robotics and Automation, 2005*(April), 661–666. <https://doi.org/10.1109/ROBOT.2005.1570193>
- Singh, A., Krause, A., Guestrin, C., & Kaiser, W. J.** (2009). Efficient Informative Sensing using Multiple Robots. *Journal of Artificial Intelligence Research, 34*, 707–755. <https://doi.org/10.1613/jair.2674>
- Stachniss, C.** (2009). *Robotic Mapping and Exploration* (Vol. 55, Issue March). Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-01097-2>
- Thrun, S., Burgard, W., & Fox, D.** (2005). *Probabilistic Robotics*. The MIT Press.
- Tokekar, P., Hook, J. Vander, Mulla, D., & Isler, V.** (2016). Sensor Planning for a Symbiotic UAV and UGV System for Precision Agriculture. *IEEE Transactions on Robotics, 32*(6), 1498–1511. <https://doi.org/10.1109/TRO.2016.2603528>
- Trujillo, J. C., Munguia, R., Guerra, E., & Grau, A.** (2018). Cooperative monocular-based SLAM for multi-UAV systems in GPS-denied environments. *Sensors (Switzerland), 18*(5), 1–24. <https://doi.org/10.3390/s18051351>
- Wilhelms, J., & Van Gelder, A.** (1992). Octrees for Faster Isosurface Generation. *ACM Transactions on Graphics (TOG), 11*(3), 201–227. <https://doi.org/10.1145/130881.130882>
- Yamauchi, B.** (1998). Frontier-based exploration using multiple robots. *Proceedings of the Second International Conference on Autonomous Agents - AGENTS '98, May*, 47–53. <https://doi.org/10.1145/280765.280773>
- Yang, S. W., & Wang, C. C.** (2011). Feasibility grids for localization and mapping in crowded urban scenes. *Proceedings - IEEE International Conference on Robotics and Automation, 2322–2328*. <https://doi.org/10.1109/ICRA.2011.5979635>
- Yoshida, K., & Tadokoro, S.** (2014). Field and Service Robotics: Results of the 8th International Conference. *Springer Tracts in Advanced Robotics, 92*(July 2012). <https://doi.org/10.1007/978-3-642-40686-7>
- Zhang, J., & Singh, S.** (2014, July 12). LOAM: Lidar Odometry and Mapping in Real-time. *Robotics: Science and Systems X*. <https://doi.org/10.15607/RSS.2014.X.007>
- Zhang, S., Wang, H., He, S., Zhang, C., & Liu, J.** (2019). An Autonomous Air-Ground Cooperative Field Surveillance System with Quadrotor UAV and Unmanned ATV Robots. *8th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, CYBER 2018, 1527–1532*. <https://doi.org/10.1109/CYBER.2018.8688331>

APPENDICES

APPENDIX A: Vehicle Models and Obstacle Avoidance

APPENDIX B: GPS Coordinate to Global Frame Transformation

APPENDIX C: Proof

APPENDIX D: Experimental Setup



APPENDIX A: Vehicle - Sensor Models and Obstacle Avoidance

In this section, the kinematic motion models of the vehicles, stochastic sensor models and obstacle avoidance methods are given.

Motion Models of Vehicles

Determining kinematic motion models is a little bit tangled process. For the wheeled vehicles, there are different types of wheel types classified as their behaviors. Tricycle drive, two-wheel differential drive, four-wheel differential drive, Ackermann drive and synchro drive are some of the most known types of vehicle motions that are allied to wheels. For the non-wheeled vehicles like submarines, quadcopters and fixed-wing aerial vehicles are also considered to extract kinematic motion models.

Further, kinematic motion models can be categorized in terms of their physical employments; as constant velocity model, holonomic model, non-holonomic model with two increments, non-holonomic model with one increment one angle sensors, black-box model and no-motion motion model (Jose Luis Blanco Claraco, 2012).

In this study, a two-wheeled differential drive robot and a quadcopter robot is used as ground and an aerial vehicle, respectively. For the ground robot, the non-holonomic with two incremental encoders kinematic model is chosen. The constant velocity kinematic model is chosen for the aerial robot.

Non-Holonomic with Two Incremental Encoders

This type of kinematic motion model is applicable for two or four-wheeled vehicles. Encoders from the differentially turned wheels obtain odometry data. Control inputs assign robots' action, and those sensors collect its feedback.

Non-holonomic motion means constrain on vehicle motions such as; the vehicle cannot move sideways as a result of robots heading must be adjusted to its velocity vector. The non-holonomic motions trajectory representation can be seen in Figure A.1.

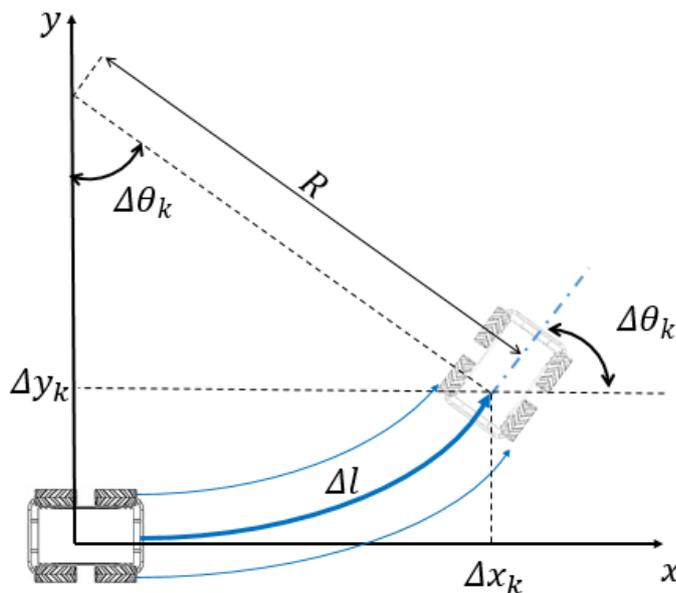


Figure A.1 : Non-holonomic motion with two-wheel encoders trajectory representation.

Robot pose at time step k is;

$$X_k = [x_k \ y_k \ \theta_k]^T \quad (\text{A.1})$$

where, x_k is the robot's position in x coordinate, y_k is the robot's position in y coordinate and θ_k is the robot's heading angle.

The control input at time step k is;

$$u_k = [\Delta u_e \ \Delta u_w]^T \quad (\text{A.2})$$

where, Δu_e means east sided wheels motion difference and Δu_w means west sided wheels motion difference.

So, within the control vector robot's states recursive function can be indicated as;

$$X_k = f(X_{k-1}, u_k) \quad (\text{A.3})$$

And it can be separated as;

$$X_k = X_{k-1} g(u_k) \quad (\text{A.4})$$

X_{k-1} states robots last pose and $g(u_k)$ is the control function which can be written as;

$$g(u_k) = [\Delta x_k \ \Delta y_k \ \Delta \theta_k]^T \quad (\text{A.5})$$

Then, by using geometric constrains $g(u_k)$ can be rewritten;

$$g(u_k) = \begin{bmatrix} \Delta x_k \\ \Delta y_k \\ \Delta \theta_k \end{bmatrix} = \begin{bmatrix} R \sin \Delta \theta_k \\ R(1 - \cos \Delta \theta_k) \\ \Delta \theta_k \end{bmatrix} \quad (\text{A.6})$$

So, the complete kinematic model of non-holonomic motion with two encoders is;

$$X_k = X_{k-1} \begin{bmatrix} R \sin \Delta \theta_k \\ R(1 - \cos \Delta \theta_k) \\ \Delta \theta_k \end{bmatrix} \quad (\text{A.7})$$

Constant Velocity Model

Constant velocity model can be used when the vehicle's odometry data could not be received from the user; thus, this type of kinematic model is suitable for quadcopters, submarines, etc. It is assumed that the robot has no acceleration at each time step; in other words, the robot's velocity is constant. In the constant velocity model state vector of the robot consists of robots pose in coordinate frame and velocity vector. In addition to this, the control action only includes time intervals namely $g(X_{k-1}, u_k)$ can be edited as $g(X_{k-1})$. Representation of the constant velocity model in coordinate frames can be seen in Figure A.2.

This model can be implemented for all vehicle types because it has no supposal on the properties of the vehicle. But, for the sake of getting a well-defined model, precise actual motion, higher sampling rates and distinct acceleration modeling are needed.

So, robot's 2D pose which is in random global coordinate frame and velocity vector can be written as;

$$X_k = [x_k \ y_k \ \theta_k \ v_{x_k} \ v_{y_k} \ v_{\theta_k}]^T \quad (\text{A.8})$$

Here, the first three elements of that matrix show robots pose in a random global coordinate frame which is denoted as R_k and the rest of the matrix indicates velocity vector v_k .

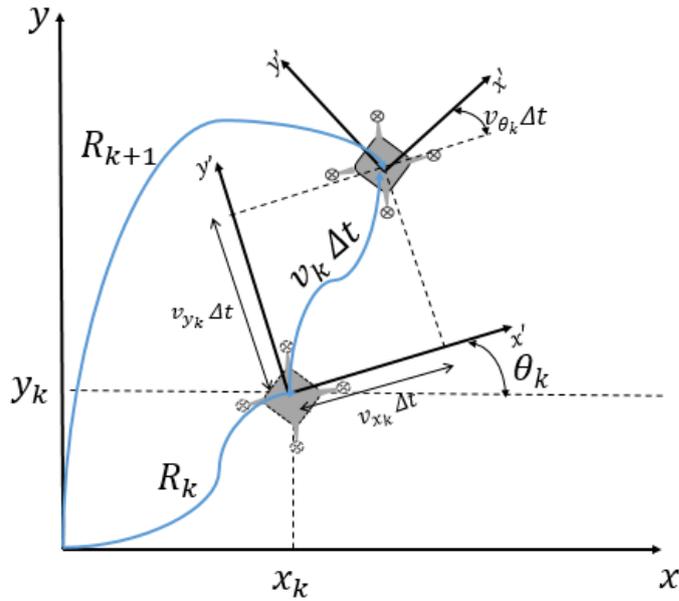


Figure A.2 : Constant velocity model representation. The dotted quadcopter indicates vehicle's position at time step k . Quadcopter without dotted lines refers to vehicle's position at time step $k + 1$.

Within the assumption that stated before as velocity is only a time interval;

$$\begin{bmatrix} R_k \\ v_k \end{bmatrix} = \begin{bmatrix} R_{k-1} \cdot (v_{k-1} \Delta t_k) \\ v_{k-1} \end{bmatrix} \quad (\text{A.9})$$

As can be seen from the equation above $v_k = v_{k-1}$ hence, the velocity vector is constant. Robot's pose can be expanded by using mathematical operations and geometric constrains in Figure A.2.

Stochastic Sensor Model

Sensors are the necessary equipment to obtain a mobile robot's position and orientation and the map of the environment where the robot is moving around. Sensor types are illustrated in Figure A.3

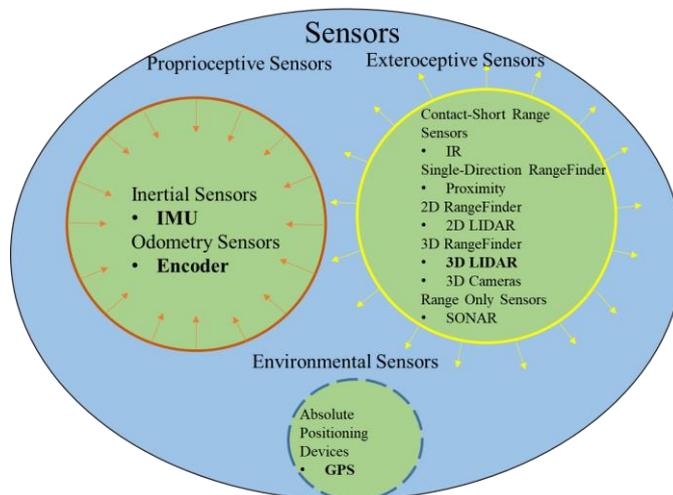


Figure A.3 : An illustrated form of sensor types and examples of products. Bolded ones are the sensors that are used in this study.

In SLAM applications, Bayesian recursive estimators employ uncertainty models of sensors. The problem remarked in this study consists of UGV, which is equipped with an IMU, two encoders located on the south and west sided wheels and GPS. Also, position UAV has its own IMU and RTK-GPS. Both UGV and UAV will be equipped with Lidar sensors.

IMU sensor is used for obtaining heading angles and inclinations in 2D on UGV's. For UAV's, this type of sensor treated collect pitch-yaw-roll angles of the vehicle. Encoders are the ones that are essential for the non-holonomic motion model. In other words, those are very important sensors for two-wheeled mobile robot applications. Further, proprioceptive sensors are employed to gather robot's state. However, it is necessary to get environmental information to localize itself and map its environment. In order to collect that information, Lidar is attained. Lidar sensors send laser beams about its 360° and collect those beams to realize the range of the reflection surface. Within that information, robots' environment can be illustrated.

Model of a sensor with additive noise can be presented as;

$$Y_k = h(x_k, m) + \vartheta_k \quad (\text{A.10})$$

Where x_k is the robot pose measured by sensor, m is the measured data of map and ϑ_k is the Gaussian sensor measurement noise with the normal distributions stated below;

$$Y_k \approx N(\bar{Y}_k, S_k) \quad (\text{A.11})$$

$$\vartheta_k \approx N(0, Q_{\vartheta_k}) \quad (\text{A.12})$$

$$m \approx N(\mu, \Sigma) \quad (\text{A.13})$$

Then the stochastic model of sensors will be;

$$\bar{Y}_k = h'(\bar{x}_k, m) + \vartheta_k \quad (\text{A.14})$$

With the covariances of IMU, encoder and GPS given as follows;

$$R^{IMU} = \begin{bmatrix} \sigma_{\varphi\varphi}^2 & \sigma_{\varphi\theta}^2 & \sigma_{\varphi\psi}^2 \\ \sigma_{\theta\varphi}^2 & \sigma_{\theta\theta}^2 & \sigma_{\theta\psi}^2 \\ \sigma_{\psi\varphi}^2 & \sigma_{\psi\theta}^2 & \sigma_{\psi\psi}^2 \end{bmatrix} \quad (\text{A.15})$$

$$R^{odom} = \begin{bmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 & \sigma_{x\psi}^2 \\ \sigma_{yx}^2 & \sigma_{yy}^2 & \sigma_{y\psi}^2 \\ \sigma_{\psi x}^2 & \sigma_{\psi y}^2 & \sigma_{\psi\psi}^2 \end{bmatrix} \quad (\text{A.16})$$

$$R^{gps} = \begin{bmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 & \sigma_{x\psi}^2 & 0 & 0 & 0 \\ \sigma_{y\varphi}^2 & \sigma_{yy}^2 & \sigma_{y\psi}^2 & 0 & 0 & 0 \\ \sigma_{\psi x}^2 & \sigma_{\psi y}^2 & \sigma_{zz}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{\dot{x}\dot{x}}^2 & \sigma_{\dot{x}\dot{y}}^2 & \sigma_{\dot{x}\dot{z}}^2 \\ 0 & 0 & 0 & \sigma_{\dot{y}\dot{x}}^2 & \sigma_{\dot{y}\dot{y}}^2 & \sigma_{\dot{y}\dot{z}}^2 \\ 0 & 0 & 0 & \sigma_{\dot{z}\dot{x}}^2 & \sigma_{\dot{z}\dot{y}}^2 & \sigma_{\dot{z}\dot{z}}^2 \end{bmatrix} \quad (\text{A.17})$$

Obstacle Avoidance

To define a collision-free path for robot obstacle avoidance must be considered for autonomous application. In this dissertation, obstacle avoidance is not a discussed point for the aerial vehicle since it is elevated over the maximum heights of the objects in the environment, and the altitude of the UAV is not changed during the process. First, the base footprint of the UGV and a tolerance value for this footprint are defined as in Figure A.4.

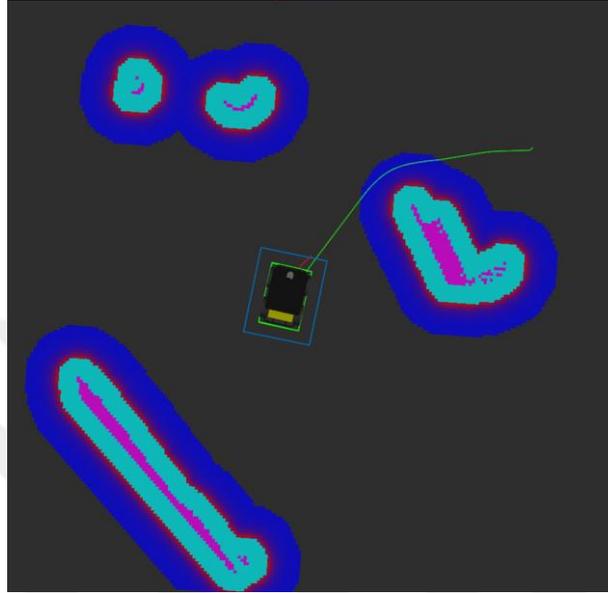


Figure A.4 : UGV obstacle avoidance features. Base footprint(green rectangle), toleranced base footprint with +0.25m longitudinal and +0.17m lateral enlargement (light blue rectangle), trajectory arc (red line), global trajectory (green line), obstacles (red boundaries, aqua section and magenta central line) and DWA velocity decreasing area (blue section).

The global path is defined as the shortest trajectory to reach the goal point. Robot heading towards the line, which connects the point of its center and the goal point, and starts moving linear in case there are not any obstacles on the path. The local path is defined with the dynamic window approach (DWA) by consideration of the frame that includes tolerances on the footprint. The DWA planner (Fox et al., 1997), provides an optimized and collision-free trajectory for the robot. An objective function, including translational and rotational velocities, is maximized to move the robot with avoiding the obstacles. The objective function of the DWA is as follows;

$$O_{DWA}(\dot{x}, \dot{\theta}) = f(\sigma. heading(\dot{x}, \dot{\theta}) + \varphi. dist(\dot{x}, \dot{\theta}) + v. vel(\dot{x}, \dot{\theta})) \quad (A.18)$$

Here, \dot{x} and $\dot{\theta}$ is translational and rotational velocity, respectively. Besides, *heading* depends on the movement of the robot, whether directly to the goal point or changing its heading angle. *dist* is a measure of the distance between the closest object and the robot, and if the distance is small, the robot's aim will become to move around it. Lastly, *vel* refers to the velocity towards the heading of the robot in order to move the target fast. Function *f*, ensures the movement with a clearance of objects through smoothening these weighted definitions.

Furthermore, the translational and rotational velocities that possibly maximize the objective function are chosen from the intersection of search spaces; trajectory arcs, acceptable velocities and dynamic window. Trajectory arcs are the possible ones that

robots can follow during the time intervals. These cannot intersect with the obstacle. Acceptable velocities refer to the bounds of velocities that do not result in a collision and will allow the robot to stop before the object. The set of acceptable velocities can be given as follows;

$$S_{V_a} = \left\{ \dot{x}, \dot{\theta} \mid \dot{x} \leq \sqrt{2 \cdot \text{dist}(\dot{x}, \dot{\theta}) \cdot \ddot{x}}, \dot{\theta} \leq \sqrt{2 \cdot \text{dist}(\dot{x}, \dot{\theta}) \cdot \ddot{\theta}} \right\} \quad (\text{A.19})$$

The dynamic window refers to the velocities that robot be able to reach in time steps; hence it is limited with regard to the vehicle's accelerations. The definition of the dynamic window velocity set is given below.

$$S_{V_d} = \left\{ \dot{x}, \dot{\theta} \mid \dot{x} \in [\dot{x}_{t-1} - \ddot{x} \cdot t, \dot{x}_{t-1} + \ddot{x} \cdot t], \dot{\theta} \in [\dot{\theta}_{t-1} - \ddot{\theta} \cdot t, \dot{\theta}_{t-1} + \ddot{\theta} \cdot t] \right\} \quad (\text{A.20})$$

Here, the t is the duration between time intervals and \dot{x}_{t-1} , $\dot{\theta}_{t-1}$ are the velocities before the control command given. The trajectory arcs outside this set of velocities are omitted since those cannot be reached.

As a result, the intersection between possible velocities that provide trajectory arcs, acceptable velocities and dynamic window ensures a local collision-free path planner.

APPENDIX B: GPS Coordinate to Global Frame Transformation

The 2D transformation between GPS coordinates to points in the global frame is done with the equations given below;

$$x = \cos(\phi) \cdot \sqrt{\frac{1}{\left(\frac{\sin(\phi)}{a}\right)^2 + \left(\frac{\cos(\phi)}{c}\right)^2}} \cdot \left[(\text{longitude2} - \text{longitude1}) \cdot \frac{\pi}{180} \right] \quad (\text{B.1})$$

and,

$$y = \sqrt{\frac{1}{\left(\frac{\sin(\phi)}{a}\right)^2 + \left(\frac{\cos(\phi)}{c}\right)^2}} \cdot \left[(\text{latitude2} - \text{latitude1}) \cdot \frac{\pi}{180} \right] \quad (\text{B.2})$$

With,

$$\phi = \frac{\pi}{2} - \frac{\text{latitude1} + \text{latitude2}}{2} \cdot \frac{\pi}{180} \quad (\text{B.3})$$

Here a is the equatorial radius of the earth and its value is 6378136.6m. Also, the polar radius of the earth is $c=6356751.9\text{m}$.

APPENDIX C: Proof of Obtaining Equation (4.9) from Equation (4.8)

Proof of obtaining equation (4.9) from equation (4.8) is given below.

First, the relation between joint probability and the conditional probability of w_s and o_s random variables as specified in below must be taken into account;

$$P(w_s, o_s) = P(w_s|o_s) P(o_s) = P(o_s|w_s) P(w_s) \quad (C.1)$$

And,

$$P(o_s) = \int_{w_s} P(o_s|w_s = w) P(w_s = w) dw_s \quad (C.2)$$

Secondly, with the chain rule of partial derivative, there is;

$$\frac{\partial P(w_s|o_s)}{\partial p_{c_i}} = \frac{\partial P(o_s|w_s) P(w_s)}{\partial p_{c_i} P(o_s)} - \frac{\partial P(o_s = o) P(o_s|w_s) P(w_s)}{\partial p_{c_i} P(o_s)^2} \quad (C.3)$$

So, replacing C.3 in equation (4.8)'s first integral operation provides;

$$\begin{aligned} \iint \frac{\partial P(w_s|o_s)}{\partial p_{c_i}} P(o_s) dw_s do_s &= \frac{\partial}{\partial p_{c_i}} \iint \partial P(o_s = o|w_s = w) P(w_s = w) dw_s do_s \\ &= \frac{\partial}{\partial p_{c_i}} 1 \\ &= 0 \end{aligned} \quad (C.4)$$

And,

$$\begin{aligned} \iint \frac{\partial P(o_s = o)}{\partial p_{c_i}} \frac{P(o_s = o|w_s = w) P(w_s = w)}{P(o_s = o)} dw_s do_s &= \int \frac{\partial P(o_s = o)}{\partial p_{c_i}} \int \frac{P(o_s = o|w_s = w) P(w_s = w)}{P(o_s = o)} dw_s do_s \\ &= \int \frac{\partial P(o_s = o)}{\partial p_{c_i}} do_s \\ &= \frac{\partial}{\partial p_{c_i}} \int P(o_s = o) do_s \\ &= \frac{\partial}{\partial p_{c_i}} 1 \\ &= 0 \end{aligned} \quad (C.5)$$

As a result of that, equation (4.8) becomes equation (4.9).

APPENDIX D: Experimental Setup

Matrice 600Pro

DJI MATRICE 600PRO is a product that has a very stable and long flight range capability, and it can carry equipment that has a lot of advanced features. This product is used in order to provide less hardware development necessity, and it is used in a lot of different research studies. As a result, it can be aimed to develop theoretical studies of this dissertation. Main properties of the UAV is stated below (Table D.1);

Table D.1 : MATRICE 600PRO main features (*Matrice 600 Pro Specs, FAQ, Tutorials and Downloads - DJI, n.d.*).

Feature	Remark
Max. Take-off weight	15.5kg
Propeller Number	6
Min. Flight Duration	16min
Max. Diameter	1668m
Weight	9.5kg
Battery Number	6
Max. Altitude	500m
Max. Meter Above Sea Level	2500m
Operating Frequency	2.4GHZ, 5.8GHz
Remote Controller	Available
Internal IMU	Available
Onboard SDK	Available

A picture of DJI MATRICE 600PRO can be seen in Figure D.1.



Figure D.1 : DJI MATRICE 600PRO.

HUSKY A200

HUSKY A200 is programmable, ROS compatible, able to carry a lot of different equipment, and a very popular vehicle in mobile robot studies. It has a 24V battery power supply, 4x4 power train, and wheels that can be used in very rough terrains. Brief features are listed in Table D.2.

Table D.2 : HUSKY A200 features (*Husky UGV - Outdoor Field Research Robot by Clearpath, n.d.*).

Feature	Remark
Max. Payload Capacity	75kg
Dimensions	990x670x390mm
Max. Duration	8h
Max. Operation Diameter	1668m
Weight	50kg
Power Output	5V / 12V / 24V, Max 5A
Control Modes	Voltage, speed, torque, wheel speed

D-RTK GPS

D-RTK GPS is provided by DJI company, which has subcentimeter accuracy of localization information. This Kinematic GPS has a ground station and onboard antennas. The features of the GPS module is given in Table D.3.

Table D.3 : D-RTK GPS features.

Feature	Remark
Location Accuracy	<1cm
Speed Accuracy	0.03m/s (RMS)
Max. Operation Time	8hours
Frequency	Global GPS L1&L2, GLONASS F1&F2
Weight	139.5g
Power Consumption	5.2W
Communication Interface	CAN, UART, USB

VLP-16 & VLP16Lite

VLP-16 and VLP16Lite are the most critical equipment that is used in this study as 3D laser measurement sensors. These equipment are located on UGV and UAV with newly designed mounting parts. Specifications of Velodyne Lidars are given in Table D.4.

Table D.4 : Velodyne VLP-16 specifications.

System	Specification	Value
	Channel Number	16
	Measurement Distance	100m
	Accuracy	±3cm
Sensor	Vertical FOV	±15°(30°)
	Horizontal FOV	360°
	Angular Resolution	0.1°-0.4°
	Rotation Rate	5-20Hz
	Beam Size	903nm
Laser	Beam Divergence	9.5mm x12.7mm
	Wavelength	0.18°
Weight	VLP16	830g
	VLP16Lite	590g

VLP16 and VLP16Lite sensors have the same outside dimensions; however, their weight differs from each other. Lite one is used as mounted on UAV and the other one is located with a certain height from the upper mounting plate on the UGV. It is desired to extend Lidar's the sight of view by omitting the dead band of the sensor measurements. Without this improvement, Lidar measures the distances to the points on the UGV.

Mounting Parts

Mounting parts are designed and manufactured with taking in consideration of general design criteria in order to mount VLP-16, VLP16Lite and their interface boxes on UAV and UGV by preventing vibration and shock effects.

Manifold

The manifold is a particularly designed high performance embedded computer that is compatible with the onboard software development kit released by DJI company. In order to create an autonomous UAV system, the manifold has a well-designed infrastructure whose technical specifications are given in Table D.5.

Table D.5 : Manifold technical specifications.

Feature	Remark
Main Processor	Quad-Core 4 Plus-1 ARM Vision Signal Processor Low Power Consumption Voice Processor Multi-time and power space
Graphic Processor	Advanced power distribution management NVIDIA Kepler GeForce
Storage	2GB DDR3L RAM 16GB eMMC 4.51 store
Voice	Microphone and Headphone Input
USB	USB 3.0, USB 2.0, Micro-B USB
Network	10/100/1000 Based Ethernet
I/O	UART, Micro SD, Mini HDMI, Mini PCIe
Weight	139.5g
Power Consumption	5.2W
Communication Interface	CAN, UART, USB



CURRICULUM VITAE



Name Surname : Mehmet Caner AKAY

Place and Date of Birth : Ankara, 21.05.1989

E-Mail : canerakay@gmail.com

EDUCATION :

- **B.Sc.** : 2010, Istanbul Technical University, Faculty of Mechanical Engineering, Department of Mechanical Engineering
- **M.Sc.** : 2012, Istanbul Technical University, Graduate School of Science Engineering and Technology, Department of Defence Technologies

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2013-2015 Mechanical Design Engineer, ALTINAY Robotics
- 2015- ... Systems Engineering Manager, ALTINAY Aerospace and Advanced Technologies

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Akay, M. C.** and Temeltaş, H., (2020). Constructing common height maps with various entropy-based similarity metrics and utilizing layering method for heterogeneous robot teams. *Industrial Robot: The International Journal of Robotics Research and Application*, 47, 8, pp.889-902. <https://doi.org/10.1108/IR-03-2020-0062>
- **Akay, M. C.**, Aybakan, A., Temeltas, H. (2018). Map Matching Performance under Various Similarity Metrics for Heterogeneous Robot Teams. *International Journal of Electrical, Electronic and Communication Sciences*, 12(12), 884–888. <https://doi.org/10.5281/zenodo.2022110>
- Aybakan, A., Haddeler, G., **Akay M. C.**, Ervan, O., Temeltas, H. (2019). A 3D LiDAR dataset of ITU heterogeneous robot team. *ACM International Conference Proceeding Series*, 12–17. <https://doi.org/10.1145/3373724.3373734>

- Haddeler, G., Aybakan, A., **Akay, M. C.**, & Temeltas, H. (2020). Evaluation of 3D LiDAR Sensor Setup for Heterogeneous Robot Team. *Journal of Intelligent & Robotic Systems*. <https://doi.org/10.1007/s10846-020-01207-y>

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- Inalhan, G., Eren, U., Baskaya, E., **Akay, M.C.** (2012). Design of a Flexible Nanosatellite Bus for Science Missions, *Proceedings of AIAA Space Conference*.
- **Akay, M.C.**, Inalhan, G. (2012). İTÜ pSAT II Tasarım ve Üretimi : Nano Uydular için Özgün Uydu Ana Sistemi ile Yönelim Belirleme ve Kontrol Sistemi Geliştirme Projesi, *6th Defense Technologies Congress (SAVTEK)*.
- Eren U., Inalhan G., **Akay, M.C.**, et. al. (2012). A Design of a Flexible Nanosatellite Bus System for ITUpSAT II, *Proceedings of International Conference on Student Small Satellites, CSSS*.
- Baskaya E., Inalhan G., **Akay M.C.**, et. al. (2012). A Precise ADCS Design for ITUpSAT II, *Proceedings of International Conference on Student Small Satellites, CSSS*.
- Koyuncu E., Cihan M., Inalhan G., **Akay M.C.**, et.al., (2011). Design and Development of ITU pSAT II: On orbit Demonstration of a High Precision ADCS for nanosatellites, *Proceedings of 8th International ESA Conference on Guidance, Navigation & Control*.
- Koyuncu E., Inalhan G., **Akay M.C.**, et.al. (2011). ITU-pSAT II :High-precision Nanosatellite ADCS Development Project, *Proceedings of Recent Advances in Space Technologies 2011,pp.500-505*.