

LIDAR Verileri ve RANSAC Algoritması Kullanılarak Otonom Mobil Robotlar İçin Hassas Yanaşma Noktası Tespiti

Melih Eren MALLI
Toprak ATEŞ
Kocaeli Üniversitesi
Bilgisayar Mühendisliği
Kocaeli, Türkiye
250201112@kocaeli.edu.tr
250201116@kocaeli.edu.tr

Özet—Bu çalışma, otonom mobil robotların şarj istasyonları gibi belirli hedeflere yüksek hassasiyetle yanaşabilmesi için kritik bir problem olan yanaşma noktası tespiti üzerine odaklanmaktadır. Problem, 2B LIDAR sensöründen elde edilen ham nokta bulutu verilerinin işlenerek, gürültü ve aykırı değerlere karşı dayanıklı bir şekilde çevresel özelliklerin çıkarılmasını gerektirmektedir. Bu amaçla, C++ programlama dili kullanılarak geliştirilen bir sistem sunulmaktadır. Sistem, özel bir TOML formatındaki sensör verilerini okuyup ön işleme adımlarından geçirdikten sonra, çevredeki doğrusal yapıları tespit etmek için RANS-AC (RANDOM SAMPLE CONSENSUS) algoritmasını temel alır. RANSAC, sensör verilerindeki kaçınılmaz gürültü ve hatalı ölçümlere rağmen güvenilir doğru modelleri üretebilme yeteneği sayesinde bu görev için ideal bir çözüm sunmaktadır. Tespit edilen doğruların geometrik analizi sonucunda, kesişim noktaları hesaplanmakta ve proje istekleri doğrultusunda önceden tanımlanmış bir açısız kısıta (kesişim açısı 60°) uyan noktalar, potansiyel yanaşma hedefi olarak belirlenmektedir. Son olarak, robotun konumu ile bu hedef nokta arasındaki mesafe hesaplanarak yanaşma manevrası için gerekli temel bilgi üretilmektedir. Bu rapor, veri işlemeden hedef tespitine kadar olan tüm metodolojiyi detaylandırmakta ve sistemin etkinliğini sağlayan test verileri üzerinde göstermektedir.

Keywords—Haritalama, Lidar, IEEE, format

I. GİRİŞ

A. Otonom Sistemlerde Hassas Konumlanmanın Önemi

Otonom Mobil Robotlar (AMR'ler), üretim hatlarından lojistik depolara, sağlık hizmetlerinden tarıma kadar geniş bir yelpazede endüstriyel süreçleri dönüştürmektedir. Bu sistemlerin verimliliği ve otonomisi, temel görevleri insan müdahalesi olmadan uzun süreler boyunca yerine getirebilme yeteneklerine doğrudan bağlıdır. Bu yeteneklerin en kritik olanlarından biri, otonom yanaşma (autonomous docking) işlemidir. Robotların şarj ünitelerine, malzeme bırakma/alma istasyonlarına veya belirli görev başlangıç noktalarına milimetrik hassasiyetle yanaşabilmesi, kesintisiz operasyonların temelini oluşturur. Hassas yanaşma olmadan, bir robotun enerji seviyesini kendi başına yönetmesi veya bir montaj hattındaki görevini doğru bir şekilde yerine getirmesi imkansız hale gelir. Bu nedenle, çevresel verileri kullanarak güvenilir ve tekrarlanabilir bir yanaşma stratejisi geliştirmek, modern robotik sistemler için temel bir araştırma ve mühendislik problemidir.

B. Çevresel Algılama için 2B LIDAR Sensörleri

Otonom robotların çevrelerini algılaması için çeşitli sensör teknolojileri kullanılmaktadır. Bunlar arasında, Işık Tespiti ve Mesafe Ölçümü (Light Detection and Ranging - LIDAR)

sensörleri, özellikle iç mekan uygulamalarında sundukları yüksek doğruluk, güvenilirlik ve farklı aydınlatma koşullarından etkilenmeme gibi avantajlarla öne çıkmaktadır. 2B LIDAR sensörleri, tipik olarak Uçuş Süresi (Time of Flight - ToF) prensibiyle çalışır. Bir lazer darbesi gönderir ve bu darbenin bir cisme çarpıp sensöre geri dönmesi için geçen süreyi ölçer. Bu süre, ışık hızı kullanılarak cismin sensöre olan mesafesine dönüştürülür. Sensörün kendi eksenini etrafında dönmesiyle, çevrenin 360 derecelik iki boyutlu bir haritası, yani bir nokta bulutu (point cloud) oluşturulur. Bu nokta bulutu, her biri bir açı ve o açıya karşılık gelen bir mesafe (range) değerinden oluşan bir dizi ölçümden meydana gelir ve robotun çevresindeki nesnelerin konturlarını temsil eder. Bu çalışma, robotun çevresindeki geometrik yapıları anlamlandırmak için birincil veri kaynağı olarak bu 2B LIDAR verilerini kullanmaktadır.

C. Nokta Bulutu Veri İşlemedeki Zorluklar: Gürültü ve Aykırı Değerler

LIDAR sensörleri yüksek doğruluk sunsa da, ürettikleri ham veriler hiçbir zaman mükemmel değildir. Gerçek dünya ortamlarında, sensör verileri çeşitli nedenlerle bozulmalara uğrayabilir. Yansıtıcı veya şeffaf yüzeyler, sensörün algılama menzili dışındaki nesneler, çevresel parazitler veya sensörün kendi içsel gürültüsü, nokta bulutunda hatalı ölçümlere yol açar. Bu hatalı ölçümler iki ana kategoriye ayrılır: gürültü (noise), yani gerçek değer etrafındaki küçük sapmalar, ve aykırı değerler (outliers) veya "kaba hatalar" (gross errors), yani modelle hiçbir ilişkisi olmayan tamamen yanlış ölçümler. Proje tanımında belirtilen -1.0 gibi geçersiz okumalar, bu aykırı değerlere bir örnektir. Çevresel özellikleri (örneğin duvarlar, raflar gibi doğrusal yapılar) tespit etmek için kullanılan geleneksel model uydurma yöntemleri, örneğin en küçük kareler (least squares) metodu, bu aykırı değerlere karşı son derece hassastır. Tek bir aykırı nokta bile, uydurulan modelin (örneğin bir doğrunun) gerçek yapıdan önemli ölçüde sapmasına neden olabilir. Bu durum, robotun çevresini yanlış yorumlamasına ve dolayısıyla yanaşma gibi hassas görevlerde başarısız olmasına yol açar.

D. RANSAC: Aykırı Değerlere Karşı Dayanıklı Bir Model Uydurma Paradigması

Yukarıda belirtilen zorlukların üstesinden gelmek için, aykırı değerlere karşı doğası gereği dayanıklı (robust) bir algoritmaya ihtiyaç duyulmaktadır. Bu noktada, 1981 yılında Fischler ve Bolles tarafından önerilen RANSAC (RANDOM SAMPLE CONSENSUS) algoritması devreye girer. RANSAC, verinin "ilintililer" (inliers), yani belirli bir matematiksel modele uyan noktalar, ve "aykırı" (outliers), yani bu modele uymayan noktalar içerdiği temel varsayımına dayanır.

Algoritma, en küçük kareler gibi tüm veri noktalarını kullanarak tek bir "optimal" model bulmaya çalışmak yerine, yinelemeli ve olasılıksal bir yaklaşım benimser. Rastgele seçilen küçük veri alt kümelerinden potansiyel modeller (hipotezler) üretir ve ardından tüm veri setinin ne kadarının bu hipotezi desteklediğini (konsensüs) kontrol eder. Bu "hipotez kur ve doğrula" (hypothesize-and-verify) mekanizması, aykırı değerlerin model tahmin süreci üzerindeki etkisini ortadan kaldırır. Bu felsefe, RANSAC'ı, hata yapmaya yatkın sensörlerden gelen verileri yorumlamak için ideal bir paradigma haline getirir. Bu projedeki temel tasarım kararı, yalnızca bir algoritma seçimi değil, aynı zamanda robotik uygulamalarda karşılaşılan veri belirsizliği sorununa yönelik felsefi bir yaklaşımdır. Amaç, matematiksel olarak mükemmel ama kırılgan bir çözüm yerine, gerçek dünya verileriyle tutarlı çalışan, "yeterince iyi" ve güvenilir bir model bulmaktır.

E. Çalışmanın Amacı ve Yapısı

Bu çalışmanın temel amacı, otonom bir mobil robotun 2B LIDAR verilerini kullanarak hassas bir yavaşlama noktasını otonom olarak tespit etmesini sağlayan bütüncül bir sistemin tasarımı, C++ ile gerçekleştirimini ve analizini sunmaktır. Bu sistem, ham sensör verilerinin okunmasından başlayarak, bu verileri anlamlı geometrik ilkelere (doğrulara) dönüştüren ve son olarak bu ilkelerden görev-odaklı bir hedef noktası (belirli bir açıya sahip kesişim noktası) çıkaran bir "soyutlama boru hattı" (abstraction pipeline) olarak tasarlanmıştır. Bu rapor, bu boru hattının her bir aşamasını detaylı bir şekilde ele almaktadır. Raporun ikinci bölümünde, sistemin mimarisi ve kullanılan yöntemler, veri ön işlemeden RANSAC tabanlı doğru tespitine ve geometrik analize kadar adım adım açıklanmaktadır. Üçüncü bölümde, geliştirilen sistemin sağlanan test veri setleri üzerindeki performansı, deneysel sonuçlar ve görselleştirmelerle birlikte sunulmakta ve değerlendirilmektedir. Dördüncü ve son bölümde ise, çalışmanın genel bir özeti sunulmakta, metodolojinin güçlü ve zayıf yönleri tartışılmakta ve gelecekte yapılabilecek potansiyel iyileştirmeler ve çalışmalar için önerilerde bulunmaktadır.

II. SİSTEM MİMARİSİ VE YÖNTEM

A. Veri Edinimi ve Ön İşleme

Bu ilk aşama, sensörden gelen ham verileri, sonraki adımlarda işlenmeye uygun, temiz ve yapılandırılmış bir formata dönüştürmeyi amaçlar. Akış şeması şu şekildedir.

<https://app.eraser.io/workspace/R5DQSOaphhrf8jMYUgrc?origin=share>

1) TOML Veri Formatının Analizi ve Ayırıştırılması

Projenin girdisi, TOML (Tom's Obvious, Minimal Language) formatında yapılandırılmış metin dosyalarıdır. Bu dosyalar, bir LIDAR taramasına ait meta verileri ve asıl ölçüm verilerini içerir. Meta veriler arasında taramanın açılma başlangıcı (`angle_min`), bitişi (`angle_max`), açılma artışı miktarı (`angle_increment`), ve sensörün minimum/maksimum algılama mesafeleri (`range_min`, `range_max`) gibi parametreler bulunur. Asıl ölçüm verileri ise, `ranges` adı verilen ve her bir açılma ışın için ölçülen mesafeleri metre cinsinden içeren bir kayan noktalı sayı dizisidir. Proje isterleri doğrultusunda, bu TOML dosyasını ayırıştırmak için herhangi bir harici kütüphane kullanılmamıştır. Bunun yerine, C++'ın standart dosya G/Ç (`fstream`) ve dize işleme (`string`, `sstream`) kütüphaneleri kullanılarak özel bir ayırıştırıcı (parser)

geliştirilmiştir. Bu ayırıştırıcı, dosyayı satır satır okur, anahtar-değer çiftlerini (örneğin, `angle_min = -1.5707963705062866`) tespit eder, = karakterine göre ayırır ve ilgili değişkenlere atama yapar. `ranges` dizisi gibi çok elemanlı veriler için, [ve] karakterleri arasındaki virgülle ayrılmış sayısal değerler okunarak bir `std::vector<float>` veri yapısında saklanır.

2) Geçersiz Verilerin Filtrelenmesi

Ham `ranges` dizisi, çeşitli nedenlerle geçersiz ölçümler içerebilir. Proje tanımında belirtildiği üzere, bu geçersiz verilerin filtrelenmesi, veri kalitesini artırmak için atılan ilk ve en önemli adımdır. Filtreleme süreci üç kurala dayanır: NaN Benzeri Değerler: Sensör bir engel algılayamadığında, `ranges` dizisine -1.0 gibi özel bir değer kaydedilir. Bu değerler, fiziksel bir mesafeyi temsil etmediği için analizden çıkarılır. Minimum Mesafe Altı Değerler: Sensörün çok yakınındaki (örneğin robotun kendi şasisi) nesnelerden gelen yansımalar gürültüye neden olabilir. Bu nedenle, `range_min` parametresinden daha küçük olan tüm ölçümler geçersiz kabul edilir ve atılır. Maksimum Mesafe Üstü Değerler: Sensörün algılama limitinin ötesindeki ölçümler veya herhangi bir yansıma alınamayan durumlar, `range_max` değerinden büyük okumalara yol açabilir. Bu ölçümler de benzer şekilde filtrelenir. Bu filtreleme adımı sonucunda, yalnızca sensörün geçerli çalışma aralığında bulunan ve fiziksel bir engeli temsil eden mesafe ölçümleri kalır.

3) Kutupsal Koordinatlardan Kartezyen Nokta Bulutuna Dönüşüm

LIDAR sensörü, verileri doğal olarak kutupsal koordinat sisteminde (mesafe r , açı θ) üretir. Ancak, geometrik analizler (doğru tespiti, kesişim hesaplama vb.) için Kartezyen koordinat sistemi (x , y) çok daha uygundur. Bu nedenle, filtrelenmiş her bir geçerli mesafe ölçümü, Kartezyen koordinatlara dönüştürülür. Dönüşüm, standart trigonometrik formüller kullanılarak gerçekleştirilir. `ranges` dizisindeki i -inci eleman olan r_i için karşılık gelen açı θ_i şu şekilde hesaplanır:

$$\theta_i = \text{angle_min} + i \times \text{angle_increment}$$

Burada i , dizideki ölçümün indeksidir (0'dan başlayarak). Bu açı ve mesafe değeri kullanılarak Kartezyen koordinatlar (x_i , y_i) aşağıdaki formüllerle bulunur:

$$x_i = r_i \times \cos \theta_i$$

$$y_i = r_i \times \sin \theta_i$$

Bu işlem, tüm geçerli ölçümler için tekrarlanır ve sonuç olarak, robotun konumunun orijin (0,0) kabul edildiği bir 2B nokta bulutu elde edilir. Bu nokta bulutu, RANSAC algoritmasının girdisini oluşturur.

B. RANSAC Tabanlı Doğru Tespiti

Bu aşama, sistemin kalbini oluşturur ve ön işlenmiş nokta bulutundan anlamlı geometrik yapılar olan doğruları çıkarmayı hedefler.

1) Algoritmanın Teorik Temelleri

RANSAC, aykırı değerlerin varlığında dahi güvenilir model parametreleri tahmin etmek için tasarlanmış, yinelemeli bir yöntemdir. Algoritmanın çalışma prensibi, bir "hipotez kur ve doğrula" döngüsüne dayanır. Doğru tespiti için bu döngü şu adımları içerir:

Hipotez Oluşturma (Hypothesis Generation): Tüm nokta bulutundan rastgele iki nokta seçilir. İki nokta, bir doğruyu

tanımlamak için gereken minimum veri setidir (minimal sample set).

Model Uydurma (Model Fitting): Bu iki noktadan geçen benzersiz doğru denklemi ($Ax + By + C = 0$) hesaplanır. Bu, o anki hipotez modelidir.

Doğrulama ve İlinti Kümesi Oluşturma (Verification and Consensus Set Formation): Nokta bulutundaki diğer tüm noktalar bu hipotez doğruya karşı test edilir. Her bir noktanın doğruya olan dik uzaklığı hesaplanır. Eğer bu uzaklık, önceden belirlenmiş bir mesafeEşiği (distanceThreshold) değerinden küçükse, o nokta "ilintili" (inlier) olarak kabul edilir ve o anki hipotezin "ilinti kümesine" (consensus set) eklenir.

Model Değerlendirme (Model Evaluation): İlinti kümesindeki nokta sayısı sayılır. Eğer bu sayı, yine önceden belirlenmiş bir minimumNoktaSayısı eşiğini (proje için 8) aşıyorsa, bu hipotez geçerli bir model olarak kabul edilir

Tekrarlama ve En İyi Modeli Seçme (Iteration and Best Model Selection): Yukarıdaki dört adım, maksimumİterasyon (max_trials) sayısı kadar tekrarlanır. Her iterasyon sonunda, en fazla sayıda ilintili noktaya sahip olan (yani en büyük ilinti kümesini oluşturan) model, o ana kadar bulunan en iyi model olarak saklanır. Döngü tamamlandığında, saklanan bu en iyi model, nokta bulutundaki en belirgin doğruyu temsil eder.

2) Uygulamaya Özel Parametrelerin Belirlenmesi

RANSAC'ın performansı, seçilen parametrelere büyük ölçüde bağlıdır ve bu parametrelerin doğru ayarlanması, algoritmanın başarısı için kritik öneme sahiptir. Bu parametreler, soyut bir algoritmayı somut bir fiziksel probleme uyarlayan köprülerdir; her biri, sensörün ve çevrenin özellikleri hakkında yapılan örtük varsayımları kodlar.

Mesafe Eşiği (distanceThreshold): Bir noktanın bir doğruya "yakın" olarak kabul edilmesi için gereken maksimum mesafeyi tanımlar. Bu değer, LIDAR sensörünün beklenen gürültü seviyesi ve ortamdaki yüzeylerin ne kadar düzgün olduğu gibi fiziksel gerçekliklere dayanarak ayarlanmalıdır. Çok küçük bir eşik, gürültü nedeniyle gerçekte doğruya ait olan noktaların dışlanmasına neden olabilir. Çok büyük bir eşik ise, aykırı değerlerin yanlışlıkla ilinti kümesine dahil edilmesine yol açarak modelin doğruluğunu bozar.

Maksimum İterasyon Sayısı (max_trials): Algoritmanın kaç farklı hipotez deneyeceğini belirler. Bu sayının artırılması, yalnızca ilintili noktalardan oluşan bir rastgele örneklem seçme olasılığını ve dolayısıyla en iyi modeli bulma ihtimalini artırır. Ancak, bu durum hesaplama maliyetini de doğru orantılı olarak yükseltir. Bu parametre, uygulamanın gerçek zamanlılık gereksinimleri ile istenen güvenilirlik seviyesi arasında bir denge kurularak seçilmelidir.

Minimum İlinti Noktası Sayısı: Bir hipotezin geçerli bir doğru olarak kabul edilmesi için gereken minimum destekçi nokta sayısıdır. Proje isterlerinde bu değer 8 olarak belirtilmiştir. Bu eşik, birkaç rastgele noktanın tesadüfen bir

doğru üzerinde hizalanmasıyla oluşan sahte modellerin elenmesine yardımcı olur.

C. Geometrik Analiz ve Hedef Belirleme

Son aşama, RANSAC tarafından tespit edilen geometrik ilkelerden (doğrulardan), görev için anlamlı olan spesifik bir hedef noktasını çıkarmaktır. Bu, sistemin genel geometrik algılamadan, görev odaklı "anlamlandırmaya" geçtiği adımdır.

1) Tespit Edilen Doğruların Kesişim Noktalarının Hesaplanması

Önceki adımda elde edilen doğru listesindeki her olası doğru çifti (L_1, L_2) için kesişim analizi yapılır. Her doğru, genel formda $A_1x + B_1y + C_1 = 0$ denklemi ile temsil edilir. İki doğrunun kesişim noktası (x_k, y_k), aşağıdaki iki bilinmeyenli lineer denklem sisteminin çözülmesiyle bulunur:

$$\begin{aligned} A_1x + B_1y &= -C_1 \\ A_2x + B_2y &= -C_2 \end{aligned}$$

Bu sistem, doğrular paralel veya çakışık değilse (yani, $A_1B_2 - A_2B_1 \neq 0$ ise) benzersiz bir çözüme sahiptir. Çözüm, Cramer kuralı veya yerine koyma metodu gibi standart cebirsel yöntemlerle hesaplanır. Bu işlem, listedeki tüm doğru çiftleri için tekrarlanarak potansiyel kesişim noktalarının bir listesi oluşturulur.

2) Açısal Kısıta Göre Geçerli Kesişimlerin Seçilmesi:

Oluşturulan kesişim noktası listesi, henüz görev için anlamlı değildir. Proje, yaşama hedefinin belirli bir geometrik özelliğe sahip olduğunu varsayar: bu hedef, aralarında en az 60 derecelik bir açı olan iki doğrunun kesişimiyle oluşur. Bu, yaşama istasyonunun belirgin bir "köşe" yapısına sahip olduğu yönündeki alan bilgisini (domain knowledge) sisteme dahil eden kritik bir adımdır. Her bir kesişim noktası için, o noktayı oluşturan iki doğrunun normal vektörleri veya eğimleri arasındaki açı hesaplanır. Eğer bu açı 60° veya daha büyükse, kesişim noktası geçerli bir yaşama hedefi adayı olarak kabul edilir. Bu filtreleme adımı, alakasız kesişimleri (örneğin, neredeyse paralel iki duvarın uzak bir noktadaki teorik kesişimi) eleyerek, hedefin ayırt edici özelliğini yakalar. Bu, sistemin "tüm doğruları bul" komutundan "yaşama istasyonumun köşesini bul" komutuna geçtiği "semantik sıçrama" anıdır.

3) Robot-Hedef Mesafesinin Belirlenmesi

Açısal filtrelemeyi geçen geçerli hedef noktası (veya noktaları) belirlendikten sonra, son adım bu hedefe olan mesafenin hesaplanmasıdır. Robotun konumu koordinat sisteminin orijini (0,0) olarak kabul edildiğinden, hedef noktası (x_{hedef}, y_{hedef})'e olan Öklid mesafesi (d) aşağıdaki formülle hesaplanır:

$$d = \sqrt{x_{hedef}^2 + y_{hedef}^2}$$

Bu mesafe, robotun yaşama manevrasını planlaması için gereken temel bilgiyi sağlar ve projenin nihai çıktılarından biridir.

III. DENEYSEL SONUÇLAR VE DEĞERLENDİRME

Sistem, C++17 standardı kullanılarak geliştirilmiş ve proje kapsamında sağlanan tarama dosyalarını kullanarak gerçek

zamanlı grafik üretir, proje tanımında belirtilen ve farklı çevresel senaryoları temsil eden beş adet TOML formatındaki veri seti (lidar1.toml, lidar2.toml, lidar3.toml, lidar4.toml, lidar5.toml) üzerinde değerlendirilmiştir. Bu veri setleri, algoritmanın farklı nokta yoğunlukları ve geometrik düzenlemeler karşısındaki tutarlılığını test edecektir. Henüz elimizde scan_data_nAn.toml dışında bir test dosyası bulunmadığından tek bir dosya üzerinden yapılan testlerden bahsedilecektir.

A. RANSAC Performans Analizi

Algoritmanın tekrarlanabilirliğini ve performansını sağlamak amacıyla, tüm test senaryolarında sabit RANSAC parametreleri kullanılmıştır. Bu parametrelerin seçimi, hem proje isterlerine hem de literatürdeki genel kabul görmüş pratiklere dayanmaktadır. Bunlar “Mesafe Eşiği, Maksimum İterasyon ve Minimum Nokta Sayısı” temel RANSAC algoritması değerleridir.

B. Veri Görselleştirme

görselleştirildiğinde, proje tanımındaki örnek çıktılarına benzer bir sonuç elde edileceği öngörülmektedir. Bu konseptsel görselleştirme şu unsurları içerecektir: Tüm Noktalar: Filtrelemeden sonra elde edilen geçerli Kartezyen noktalar, genellikle gri veya siyah küçük daireler olarak çizilir. Bu, robotun algıladığı çevrenin genel bir haritasını sunar. Robot Konumu: Robot, koordinat sisteminin merkezi olan (0,0) noktasına, örneğin mavi renkli büyük bir daire ile yerleştirilir. Tespit Edilen Doğrular ve İlintili Noktalar: RANSAC algoritması tarafından tespit edilen her bir doğru, farklı bir renkle temsil edilir. Örneğin, birinci doğru (d1) ve ona ait ilintili noktalar kırmızı, ikinci doğru (d2) ve noktaları yeşil olarak gösterilir. Bu, algoritmanın hangi noktaları hangi geometrik yapıya atadığını net bir şekilde ortaya koyar. Geçerli Kesişim Noktası: 60° veya daha büyük bir açıyla kesişen doğruların oluşturduğu hedef noktası, belirgin bir şekilde (örneğin kırmızı bir yıldız veya çarpı işareti ile) işaretlenir. Mesafe Çizgisi: Robotun merkezinden (0,0) tespit edilen hedef noktasına düz bir çizgi çizilir ve bu çizginin üzerine, hesaplanan Öklid mesafesi (örneğin, "1.90m") yazılır. Bu tür bir görsel çıktı, algoritmanın her aşamasının

sonuçlarını sezgisel olarak anlama ve doğruluğunu teyit etme açısından kritik öneme sahiptir.

IV. SONUÇ VE GELECEK ÇALIŞMALAR

Bu çalışma, 2B LIDAR verilerinden otonom robotlar için yavaş yavaş hedefi tespit etme problemini başarıyla ele almıştır. Geliştirilen C++ tabanlı sistem, RANSAC algoritmasının gürültüye ve aykırı değerlere karşı dayanıklılığından faydalanarak, ham sensör verilerini eyleme geçirilebilir bir hedefe dönüştürme yeteneğini kanıtlamıştır. Metodolojinin en belirgin gücü RANSAC'ın bu dayanıklılığı iken, en önemli sınırlılığı ise hedefin geometrisine dair sabit kodlanmış varsayımlara (doğrusal yapılar ve 60° kesişim açısı) dayanmasıdır. Bu durum, sistemin esnekliğini kısıtlamaktadır.

Gelecekteki çalışmalar, bu sistemin yeteneklerini daha da ileri taşıyabilir. Potansiyel iyileştirmeler arasında; model hassasiyetini artırmak için LO-RANSAC gibi daha gelişmiş RANSAC varyantlarının kullanılması ; sistemi daire veya diğer karmaşık şekilleri tanıyacak şekilde genişleterek geometrik esnekliği artırma ; algılama modülünü bir hareket planlama sistemiyle entegre ederek fiziksel bir robot üzerinde test etme ; ve güvenilirliği artırmak için LIDAR verilerini kamera gibi diğer sensörlerle birleştirme (sensör füzyonu) bulunmaktadır.

REFERENCES

- [1] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, June 1981.
- [2] O. Chum and J. Matas, "Locally Optimized RANSAC," in *Pattern Recognition, 25th DAGM Symposium, Proceedings*, Magdeburg, Germany, September 10-12, 2003, pp. 223–233.
- [3] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, 2007.
- [4] D. Scaramuzza and F. Fraundorfer, "Visual Odometry: Part I - The First 30 Years and Fundamentals," *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 80-92, Dec. 2011.
- [5] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.
- [6] Introduction To Algorithms Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein 3rd edition. Ch 33.4

