MARMARA ÜNİVERSİTESİ  
Teknik Bilimler Meslek Yüksekokulu

AirSim ile Unreal Engine Tabanlı   
Görüntü İşleme Destekli Otonom Drone

Bitirme Projesi

Berk Efe Polat - 360123012

Minel Baştor - 360123018

Serra Çınay Tezer - 360123515

Melike Can - 360123060

**Bölüm: Bilgisayar Teknolojileri**

**Program: Bilgisayar Programcılığı**

Danışman: Ercan Erkalkan

19/06/2025

**Özgünlük Bildirisi**

1. Bu çalışmada, başka kaynaklardan yapılan tüm alıntıların, ilgili kaynaklar referans gösterilerek açıkça belirtildiğini,
2. Alıntılar dışındaki bölümlerin, özellikle projenin ana konusunu oluşturan teorik çalışmaların ve yazılım/donanımın bizim tarafımızdan yapıldığını bildiririm.

AirSim ile Unreal Engine Tabanlı   
Görüntü İşleme Destekli Otonom Drone

(ÖZET)

Bu bitirme projesinin temel amacı, simülasyon ortamında çalışabilen, görüntü işleme ve derinlik verisi analizi kullanarak çevresel engellerden otonom olarak kaçınabilen çoklu drone sisteminin tasarımı ve gerçekleştirilmesidir. Otonom hava araçları, günümüzde tarım, lojistik, savunma sanayi ve sivil uygulamalar gibi birçok alanda yaygın olarak kullanılmakta olup, çevre algısı ve karar verme mekanizmalarının yazılım temelli olarak geliştirilmesi, bu sistemlerin sürdürülebilirliği ve erişilebilirliği açısından kritik öneme sahiptir.

Proje kapsamında, **Unreal Engine 4.27** simülasyon motoru ve **Microsoft AirSim** eklentisi kullanılarak ormanlık bir araziyi temsil eden sanal bir test ortamı oluşturulmuştur. Geliştirilen sistem, **Python programlama dili** ile yazılmış otonom kontrol algoritmalarına dayanmakta olup, drone'lar; simülasyon içerisindeki ön kamera görüntüleri ve derinlik haritalarından elde edilen verileri işleyerek çevresel engelleri tespit edebilmekte ve bu doğrultuda yön tayini, yükseklik güncellemesi veya yön değiştirme gibi manevraları gerçekleştirebilmektedir.

Sistem, yalnızca yazılım temelli çevre algılama yaklaşımı ile çalışmakta; harici sensör ya da ek donanım kullanımı gerektirmemektedir. Görsel veriler üzerinden **OpenCV kütüphanesi** ile yapılan yoğunluk analizleri ve **AirSim derinlik verisi** yardımıyla, çevresel nesnelerin konumları tespit edilerek, çarpışmadan kaçınma algoritması dinamik olarak güncellenmiştir.

Gerçekleştirilen test senaryolarında, drone'lar farklı türdeki engeller (ağaçlar, kayalar, taşlar) ile dolu kompleks bir ortamda hedef konuma başarıyla ulaşmış; çarpışmasız otonom uçuş fonksiyonları istenilen şekilde gerçekleştirilmiştir. Bu çalışma sonucunda, düşük donanım gereksinimli ve tamamen yazılım tabanlı bir otonom çoklu drone sistemi başarıyla modellenmiş ve simülasyon ortamında doğrulanmıştır.

İçindekiler

* Özet ............................................................. 3
* Giriş ............................................................. 5
* Projenin Tanımı ve Planı ............................. 6
* Kuramsal Bilgiler ........................................ 9
* Tasarım ve Gerçekleme ............................... 12
* Sonuç ve Öneriler ........................................ 26
* Kaynaklar .................................................... 28
* Özgeçmiş ..................................................... 29

GİRİŞ

Son yıllarda İnsansız Hava Araçları (İHA), özellikle keşif, haritalama, güvenlik ve arama-kurtarma gibi kritik uygulama alanlarında yaygın bir şekilde kullanılmaktadır. Bu tür görevlerin artan karmaşıklığı, İHA'ların çevresel algı yeteneklerinin geliştirilmesini ve dinamik ortamlarda otonom hareket kabiliyeti kazanmasını gerekli kılmaktadır. Özellikle engellerle dolu, yapılandırılmamış ortamlarda çarpışmadan kaçınarak hedefe ulaşabilen sistemlerin tasarımı, güncel araştırma ve geliştirme çalışmalarının odak noktalarından biri haline gelmiştir.

Bu proje kapsamında, **Unreal Engine 4.27** tabanlı bir simülasyon ortamında çalışan **Microsoft AirSim** eklentisi kullanılarak, çoklu drone yapısına sahip bir sistemin otonom uçuş kabiliyeti simüle edilmiştir. **Python diliyle geliştirilen kontrol algoritmaları**, drone'ların yalnızca ön kamera görüntüsü ve derinlik verilerini kullanarak çevrelerini algılamasına, olası engelleri tespit etmesine ve bu engellerden kaçınmak üzere uygun manevraları gerçekleştirmesine olanak tanımaktadır.

Uygulanan algoritmalar, görsel sahne yoğunluğu ve derinlik haritalarını analiz ederek engel tespiti yapmakta; engelin pozisyonuna göre yön değiştirme veya yükseklik ayarlama stratejileri oluşturarak çarpışmalardan kaçınma davranışı sergilemektedir. Böylelikle drone’lar, önceden tanımlı bir hedefe, çevresel engelleri dinamik olarak aşarak ulaşabilmektedir.

Bu çalışma, yazılım temelli görüntü işleme ve derinlik analizi kullanılarak geliştirilen otonom kontrol algoritmalarının, çoklu drone sistemlerine entegre edilmesini amaçlamaktadır. Proje; **otonom karar alma süreçleri**, **çoklu İHA koordinasyonu**, **simülasyon tabanlı test ortamları**, **gerçek-zamanlı çevre algısı** gibi konulara katkı sağlayarak, gerçek dünya uygulamalarına yönelik algoritmalar için temel bir prototip sunmaktadır.

PROJENİN TANIMI ve PLANI

Bu proje, engellerle dolu bir ortamda birden fazla insansız hava aracının (İHA) otonom olarak uçuş gerçekleştirmesini sağlayan yazılım tabanlı bir sistemin tasarlanmasını ve bu sistemin simülasyon ortamında test edilmesini amaçlamaktadır. Simülasyon altyapısı, **Unreal Engine 4.27** oyun motoru üzerinde çalışan **Microsoft AirSim** eklentisi kullanılarak oluşturulmuştur. Geliştirilen sistem, yalnızca kamera görüntüsü ve derinlik verisi ile çevresel farkındalık sağlayarak, İHA’ların çevresindeki engelleri algılamasına ve bu engellere göre uygun manevralarla güvenli uçuş rotalarını otonom biçimde belirlemesine olanak tanımaktadır.

Proje, dört kişilik bir ekip tarafından yürütülmüş olup, yazılım geliştirme, görüntü işleme algoritmalarının tasarımı, test süreçlerinin yürütülmesi ve teknik dokümantasyonun hazırlanması gibi görevler ekip üyeleri arasında iş bölümü yapılarak koordine edilmiştir.

#### **Projenin Temel Hedefleri**

* Görüntü işleme ve derinlik verisi analizine dayalı bir **engel algılama algoritması** geliştirmek.
* Birden fazla drone’un **otonom yön değiştirme** ve **yükseklik ayarlama** stratejileriyle çevresel engellerden kaçınmasını sağlamak.
* **Python programlama dili** ve **AirSim API** aracılığıyla drone’ların uçuş kontrolünü sağlamak.
* Simülasyon ortamında farklı test senaryoları yürüterek sistemin **doğruluk**, **kararlılık** ve **güvenlik** performanslarını analiz etmek.

Bu proje ile, gerçek dünya koşullarında uygulanabilir nitelikte, düşük donanım maliyetli ve yazılım temelli bir otonom drone kontrol sistemi prototipi geliştirilmiş ve başarıyla doğrulanmıştır.

Proje Planı:

Proje çalışması üç ana aşamada yürütülmüştür:

**Hazırlık ve Araştırma Aşaması**

Bu aşamada, simülasyon ortamının altyapısı oluşturulmuştur.

* **Microsoft AirSim eklentisi**, **Unreal Engine 4.27** üzerine entegre edilerek sanal test sahası yapılandırılmıştır.
* **Python tabanlı kontrol modülü** geliştirilmiş; temel uçuş komutları (kalkış, irtifa kontrolü, yön tayini) başarıyla uygulanarak sistemin başlangıç testleri gerçekleştirilmiştir.

#### **Algoritma Geliştirme Aşaması**

Bu aşamada, drone’ların çevresel farkındalığını sağlayan karar mekanizmaları geliştirilmiştir.

* **OpenCV kütüphanesi** ile görüntü işleme algoritmaları oluşturulmuş; renk ve yoğunluk temelli engel tespiti gerçekleştirilmiştir.
* **AirSim derinlik verisi** ile entegre çalışan, otonom yön ve yükseklik değiştirme kararlarını yöneten bir kontrol algoritması tasarlanmıştır.

**Test ve Sonuçlandırma Aşaması**

Geliştirilen sistem, farklı çevresel senaryolar altında kapsamlı biçimde test edilmiştir.

* Birden fazla drone’un eş zamanlı olarak uçuş gerçekleştirdiği durumlarda, engellerden çarpışmasız kaçınarak hedef konumlara ulaştığı gözlemlenmiştir.
* Algoritmanın kararlılığı, güvenliği ve doğruluğu testler sonucunda başarıyla doğrulanmıştır.

Bu üç aşamalı yapı sayesinde proje, hem yazılım odaklı akademik araştırmalara katkı sağlayabilecek düzeyde, hem de gerçek dünya uygulamalarına entegre edilebilecek potansiyelde, **uyarlanabilir bir otonom sistem prototipi** sunmaktadır.

KURAMSAL BİLGİLER

Bu projede geliştirilen otonom drone sisteminin temelini; **görüntü işleme teknikleri**, **derinlik verisi analizi** ve **AirSim simülasyon platformu** oluşturmaktadır. Aşağıda sistemin üzerine inşa edildiği temel kavramlar özetlenmiştir:

**Görüntü İşleme**

Görüntü işleme, dijital görüntülerden anlamlı bilgi çıkartılmasını sağlayan bilgisayarla görme alanının bir alt disiplinidir. Bu projede, drone’un ön kamerasından elde edilen gerçek zamanlı görüntüler üzerinde analizler yapılmıştır. **OpenCV kütüphanesi** kullanılarak görüntülerdeki **piksel yoğunluğu**, **renk dağılımı** ve **kontrast varyasyonları** incelenmiş; bu analizler aracılığıyla çevredeki engellerin tespiti gerçekleştirilmiştir. Görsel farkındalık, otonom karar mekanizmasının ilk adımını oluşturmuştur.

**Derinlik Verisi Kullanımı**

AirSim platformu, kamera görüntülerine ek olarak **derinlik haritaları** sağlamaktadır. Derinlik verisi, her pikselin drone’a olan mesafesini temsil eder ve bu sayede drone’un yakın çevresinde yer alan objelerin mesafesi hassas şekilde belirlenebilmektedir. Bu veriler, drone’un **yön tayini**, **yükseklik ayarı** ve **çarpışma önleme kararları** için temel girdi olarak kullanılmıştır. Görüntü işleme ile birleştirildiğinde, ortamın 3B yapısı daha etkin şekilde algılanabilmiştir.

**Otonom Engel Kaçınma Algoritmaları**

Drone’un hedefe yönelirken çevredeki engelleri algılayıp uygun tepkiyi verebilmesi için bir **engel algılama ve kaçınma algoritması** tasarlanmıştır. Bu algoritma;

* Engelin konumunu ve uzaklığını belirler. Belirli bir yönde çarpışma riski tespit ederse, alternatif bir rota oluşturur (örneğin sağa/sola sapma veya yukarı yönlü yükselme), böylece otonom bir şekilde karar vererek güvenli uçuş sağlar. Bu yapı, tamamen yazılım temelli bir otonom manevra sistemini mümkün kılmıştır.

**AirSim ve Unreal Engine Kullanımı**

**AirSim**, Microsoft tarafından geliştirilen ve otonom sistemler için gerçekçi fizik, sensör simülasyonu ve ortam etkileşimi sunan açık kaynaklı bir simülasyon platformudur. Bu proje kapsamında AirSim, **çoklu drone desteği**, **kamera/derinlik sensörü modellemesi** ve **Python API erişimi** gibi özellikleriyle kullanılmıştır.

AirSim, **Unreal Engine 4.27** üzerinde çalıştırılarak yüksek kaliteli 3B çevresel modelleme ve gerçek zamanlı fizik hesaplamalarının yapılması sağlanmıştır. Unreal Engine’in gelişmiş görsel motoru sayesinde, drone’ların ormanlık ve doğal engeller içeren bir arazide test edilmesi mümkün olmuştur.

****Şekil.1

TASARIM ve GERÇEKLEME

Bu projede geliştirilen sistem, **Unreal Engine 4.27** simülasyon altyapısı üzerinde çalışan **Microsoft AirSim** eklentisi ile entegre edilmiş, **Python programlama dili** kullanılarak kontrol edilen bir otonom drone modeline dayanmaktadır. Sistem; uçuş yönetimi, çevresel algılama, görüntü işleme ve karar alma olmak üzere dört temel bileşenden oluşmaktadır. Her modül, sistemin otonom ve güvenli şekilde hareket etmesini sağlamak amacıyla birbirini tamamlayan işlevler üstlenmiştir.

**1. Sistem Mimarisi**

Geliştirilen sistem dört ana modülden oluşmaktadır:

**Uçuş Kontrol Modülü**

* Bu modül, drone’un temel uçuş davranışlarını yönetir.
* **Kalkış ve iniş işlemleri**,
* **Belirli bir yükseklikte sabitlenme**,
* **Yönelme ve yön güncelleme**,
* **Konum bilgisi takibi ve kontrolü**,  
  gibi işlemler bu modül tarafından yürütülmektedir. AirSim API üzerinden alınan telemetrik veriler ile uçuş kararlılığı sağlanmaktadır.

**Hedef Takip Modülü**

* Drone’un önceden belirlenmiş bir hedefe doğru yönelmesini sağlar.
* Drone, uçuş boyunca sürekli olarak **hedef konum ile kendi pozisyonunu karşılaştırır**
* Gerekli durumlarda **yön ve hız güncellemeleri** gerçekleştirerek hedefe doğru ilerlemeye devam eder.

**Görüntü İşleme Modülü**

* Bu modül, drone’un ön kamerasından alınan görsel verileri işler.
* **OpenCV kütüphanesi** kullanılarak piksel yoğunluğu, renk dağılımı ve kontrast analizi yapılır.
* Görsel verilerden elde edilen bilgiler sayesinde **çevresel engellerin tespiti** gerçekleştirilir.
* Görüntü işleme çıktıları, engelden kaçma stratejilerine veri sağlar.

**Derinlik Tespiti ve Engel Kaçınma Modülü**

* Bu modül, AirSim tarafından sağlanan derinlik haritalarını analiz eder.
* Derinlik verileri kullanılarak, **engellerin drone’a olan mesafesi hesaplanır**.
* Eğer bir çarpışma riski algılanırsa, sistem otonom olarak:
  + **Yön değiştirerek**,
  + **Yükseklik artırarak**,  
     uygun kaçınma manevralarını gerçekleştirir.
* Böylece güvenli uçuş rotası gerçek zamanlı olarak güncellenir.
* Bu mimari yapı sayesinde, sistem yalnızca yazılımsal verilerle (kamera görüntüsü ve derinlik haritası) çevreyi algılayarak, insan müdahalesi olmadan otonom uçuş görevlerini başarıyla yerine getirebilmektedir.

1. Kod Modüllerine Dair Açıklamalar

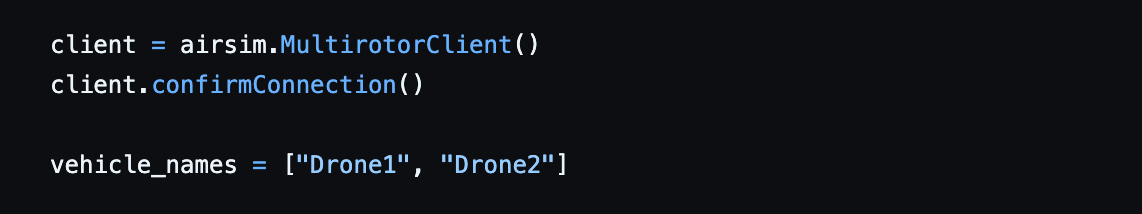
**Kod Parçası 1 – Gerekli Kütüphaneler**

  
Şekil.2

**Açıklama:**  
 Proje boyunca drone kontrolü, görüntü işleme ve matematiksel hesaplamalar için bu kütüphaneler kullanılmıştır.

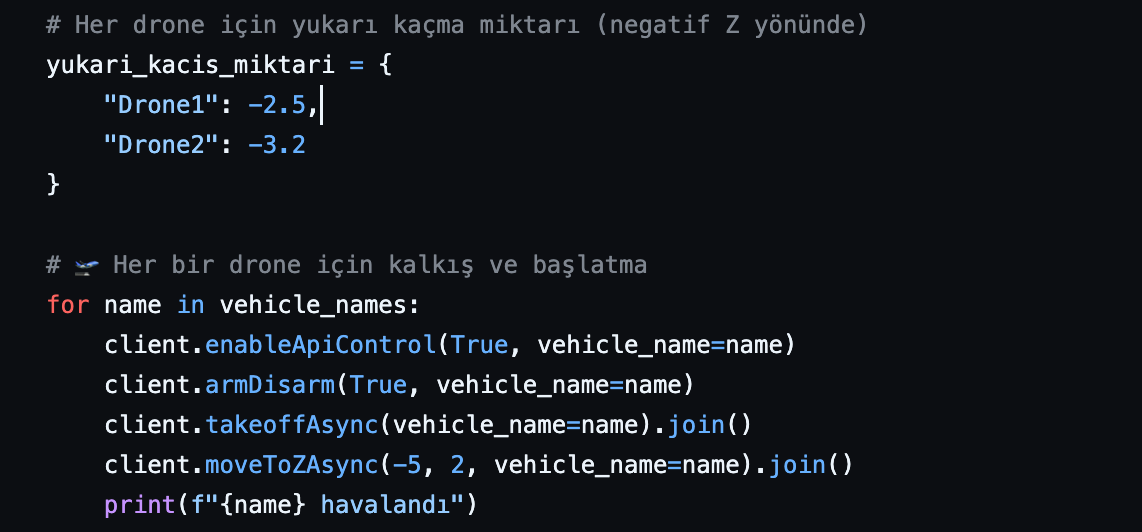
* airsim: Unreal Engine'de drone simülasyonu için API sağlar.
* numpy: Sayısal hesaplamalar için.
* cv2: Görüntü işleme (OpenCV).
* time ve math: Zamanlama ve trigonometrik hesaplamalar.

**Kod Parçası 2 – Drone Bağlantısı ve Tanımı**

  
Şekil.3

**Açıklama:**  
 AirSim istemcisi başlatılır ve simülasyon ortamına bağlanılır.  
 İki drone tanımlanır: "Drone1" ve "Drone2".

**Kod Parçası 3 – Kalkış ve Başlatma**

  
Şekil.4

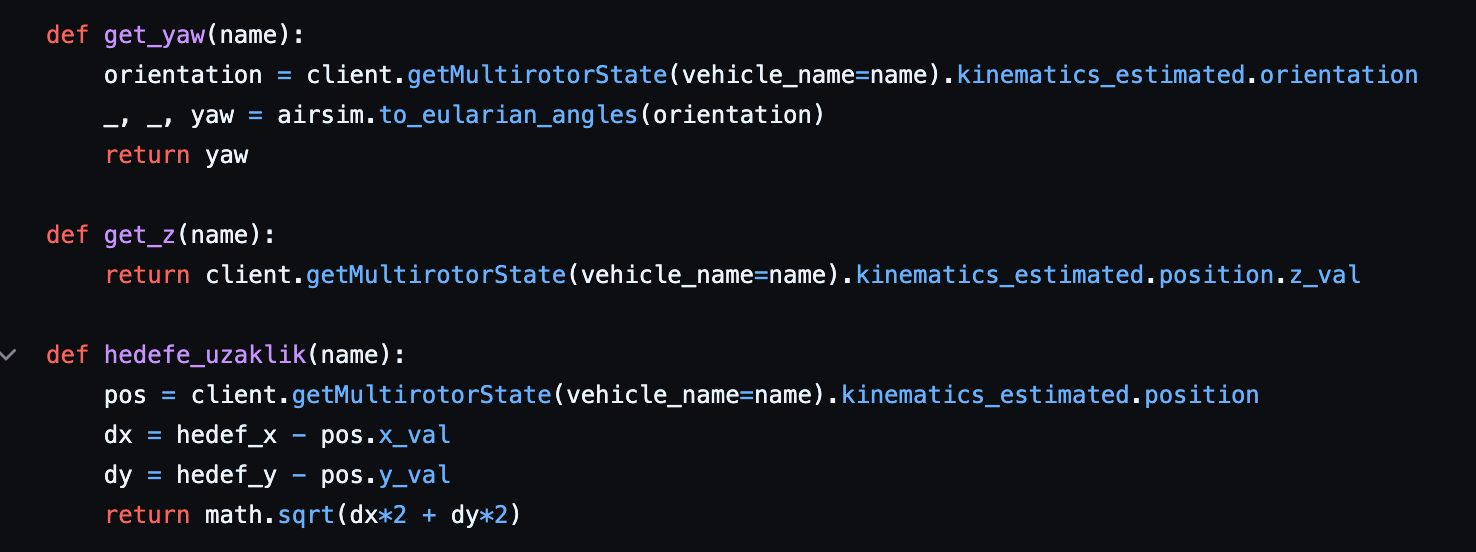
**Açıklama:**  
 Her drone için API kontrolü aktif edilir, motorları çalıştırılır ve -5 metre yüksekliğe çıkması sağlanır.  
 Yukarı kaçma hareketi için her drone’a özel Z ekseni miktarı atanmıştır.

**Kod Parçası 4 – Hedef Belirleme**

  
Şekil.5

**Açıklama:**  
 Drone'ların ulaşması gereken hedef nokta X = 30, Y = 40 olarak tanımlanmıştır.

**Kod Parçası 5 – Yardımcı Fonksiyonlar: Yön, Yükseklik, Uzaklık**

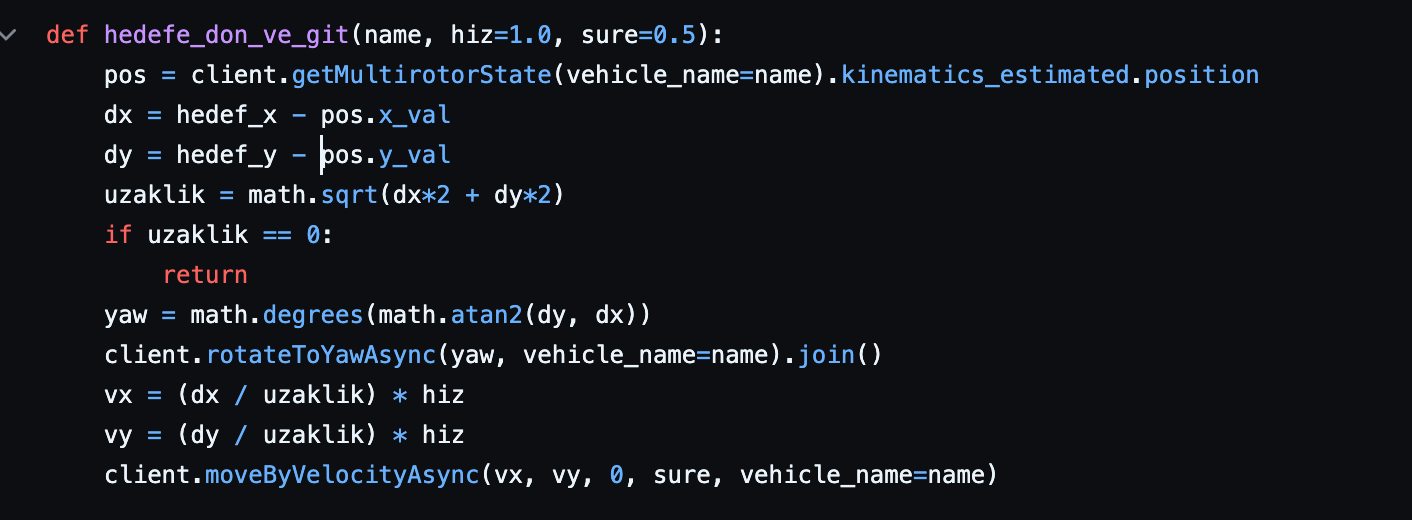
  
Şekil.6

**Açıklama:**

Bu fonksiyonlar, drone’un yönünü, yüksekliğini ve hedefe uzaklığını öğrenmek için kullanılır ve otonom hareketlerin temelini oluşturur.

* get\_yaw: Drone'un yönünü (yaw açısı) döndürür.
* get\_z: Drone’un Z eksenindeki konumunu verir.
* hedefe\_uzaklik: Drone ile hedef arasındaki mesafeyi hesaplar (Euclidean mesafe).

**Kod Parçası 6 – Hedefe Doğru Gitme**

  
Şekil.7

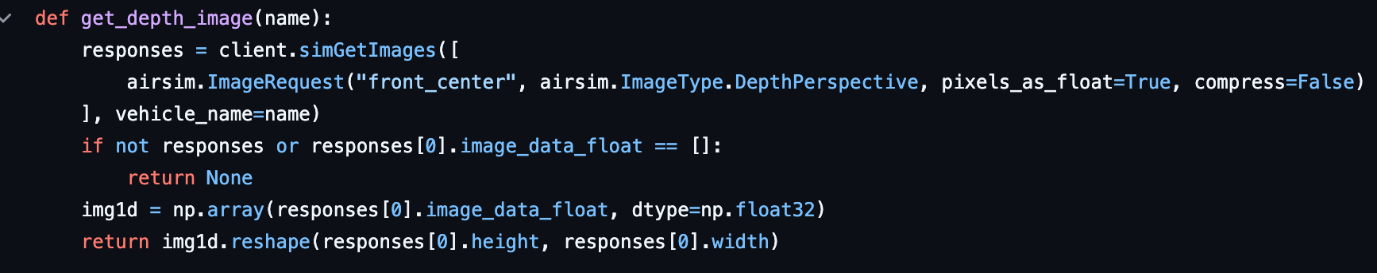
**Açıklama:**  
 Drone’un hedefe yönelip belirli hız ve süreyle ileri hareket etmesini sağlar.  
 Dönüş açısı atan2 ile hesaplanır ve drone bu açıda döndürülür, ardından sabit hızla ilerletilir.

**Kod Parçası 7 – Kamera Görüntüsü Alma**

Şekil.8

**Açıklama:**  
 Drone’un öndeki kamerasından renkli görüntü alınır. Bu görüntü daha sonra engel analizi için kullanılır.

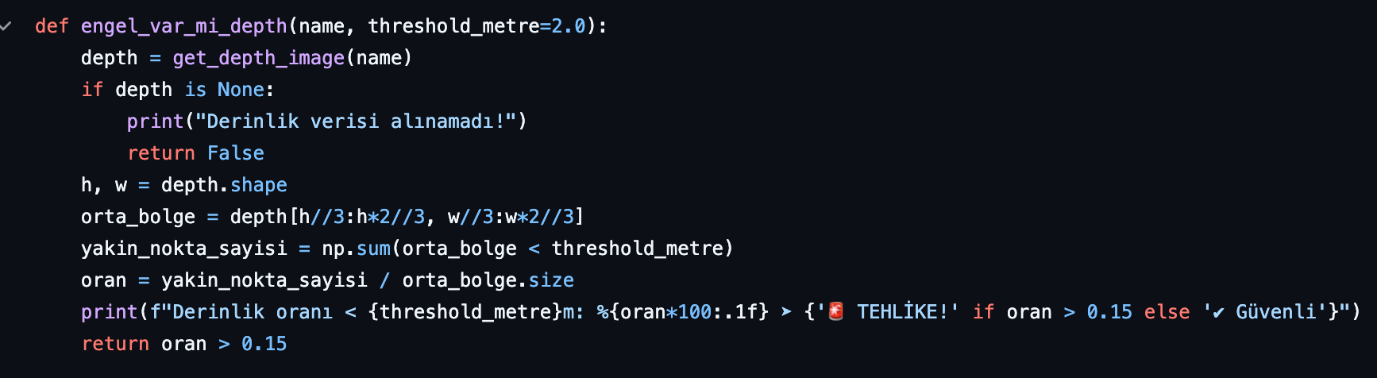
**Kod Parçası 8 – Derinlik Görüntüsü Alma**



Şekil.9

**Açıklama:**  
 Drone’un derinlik kamerasından (DepthPerspective) görüntü alınır. Bu görüntü her pikselin drone’a olan uzaklığını içerir ve metre cinsindedir.

**Kod Parçası 9 – Derinlik Tabanlı Engel Tespiti**

Şekil.10

**Açıklama:**  
 Orta bölgedeki derinlik verisi incelenir. Eğer çok sayıda piksel 2 metreden daha yakınsa, bu bir engel olarak değerlendirilir.  
 Özellikle %15’ten fazlası yakınsa “tehlikeli bölge” sayılır.

**Kod Parçası 10 – Görüntü Tabanlı Engel Puanlama**

Şekil.11

**Açıklama:**

Drone bu puanlara göre önünde ne kadar engel olduğunu, engelin yönünü ve kaçınma ihtiyacını analiz eder.

* Görüntü 4 bölgeye ayrılır: orta, sağ, sol ve alt.
* Her bölge gri tona çevrilir (cv2.cvtColor).
* Dinamik bir eşik değeri hesaplanır ve bölgeler maskeleme ile siyah-beyaz hale getirilir.
* Alt bölgede Canny ile kenar tespiti yapılır (kaya tespiti için).
* Bu şekilde engelin nerede yoğunlaştığı belirlenir.

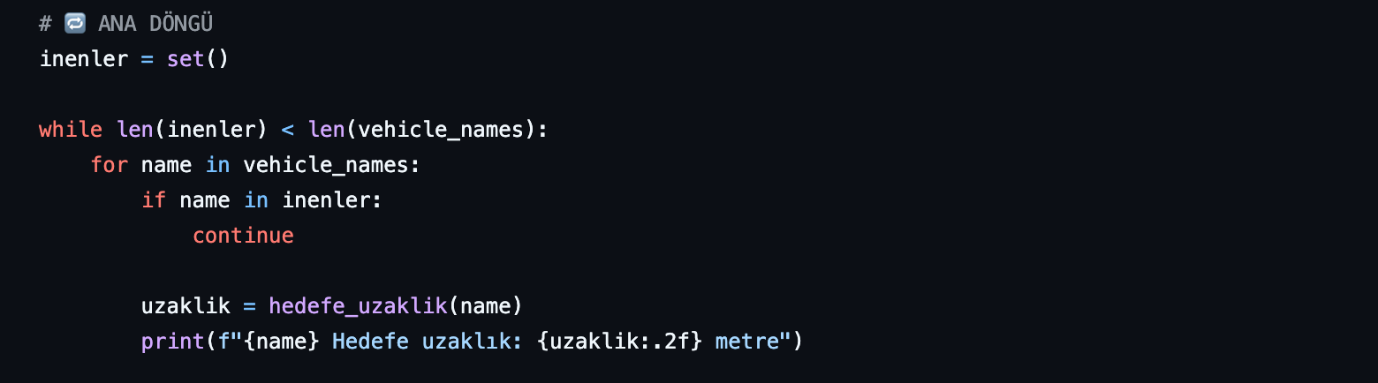
**Kod Parçası 11 – Alternatif Yön Arama**

Şekil.12

**Açıklama:**  
Drone’un önünde engel varsa, farklı yönlere dönerek hangi yönde daha az engel olduğunu bulmak. Böylece engelden kaçmak için en güvenli yön (yaw açısı) seçilir.

* Drone’un mevcut yönü (yaw) alınır.
* -60° ile +60° arasında 30’ar derecelik aralıklarla 5 farklı açı denenir (Yani: -60, -30, 0, +30, +60)
* Her açıda drone o yöne döndürülür (rotateToYawAsync).
* Yeni açıda kamera görüntüsü alınır. Görüntü alınamazsa o açı atlanır.
* Her açı için ortadaki engel yoğunluğu (puan\_orta) hesaplanır.
* En az puana (yani en az engele) sahip yön en iyi yön olarak seçilir.

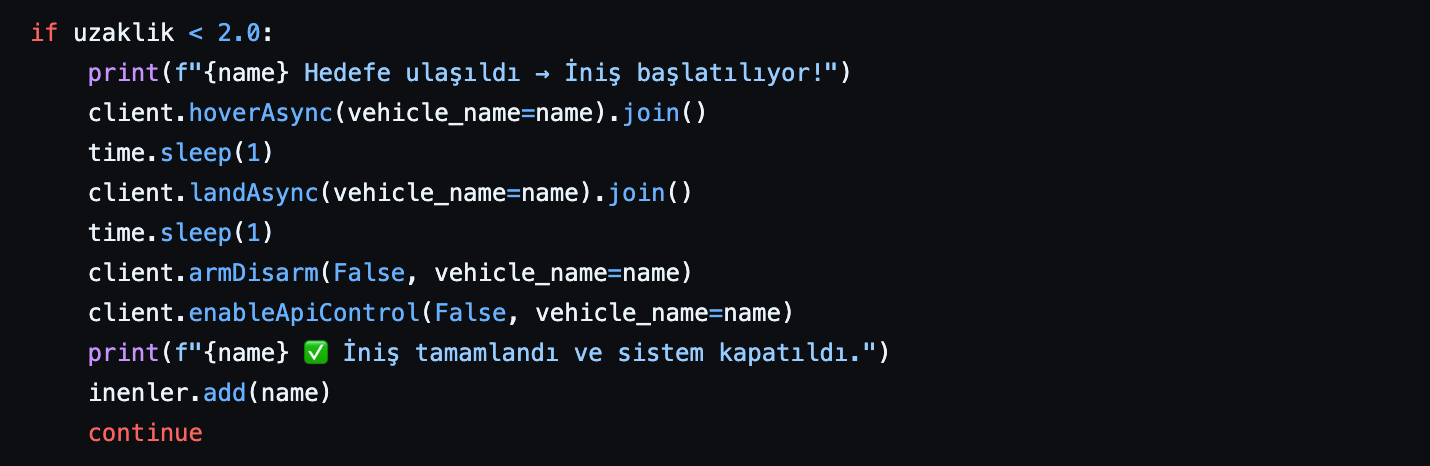
**Kod Parçası 12 – Ana Kontrol Döngüsü**

Şekil.13

**Açıklama:**  
 Bu döngü, tüm drone’lar hedefe ulaşana kadar devam eder.  
 Her adımda:

* Hedefe ulaşan drone iniş yapar.
* Derinlik ve görsel engel kontrolü yapılır.
* Engel varsa yukarı kaçılır ya da yön değiştirilir.
* Engel yoksa hedefe doğru ilerlenir.

**Kod Parçası 13 – İniş ve Sistem Kapatma**

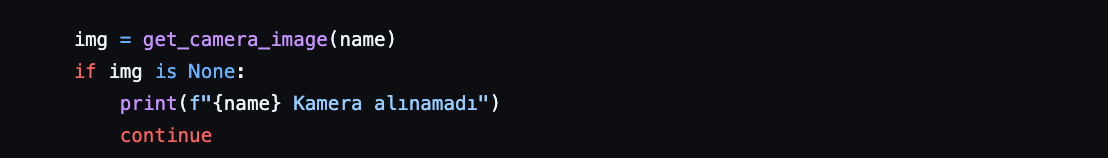
Şekil.14

**Açıklama:**  
 Drone hedefe 2 metreden fazla yaklaşırsa:

* Sabitlenir (hoverAsync)
* İniş yapar (landAsync)
* Motorları durdurur ve kontrol bırakılır.

**Kod Parçası 14 –** Engel Algılama ve Engelden Kaçınma Karar Mekanizması

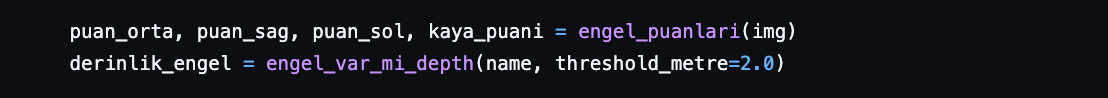
A) Kamera görüntüsü alınır:

  
Şekil.15

Açıklama:

* Ön kamera görüntüsü alınır. Eğer başarısız olursa, bu döngü turu atlanır.

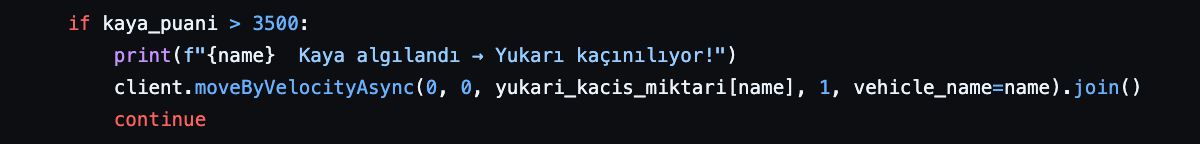
B) Görsel ve derinlik engel puanları hesaplanır:

  
Şekil.16

Açıklama:

* Görüntüdeki engellerin yönlere göre yoğunluğu hesaplanır.
* Derinlik verisine göre 2 metreden daha yakın engel olup olmadığı kontrol edilir.

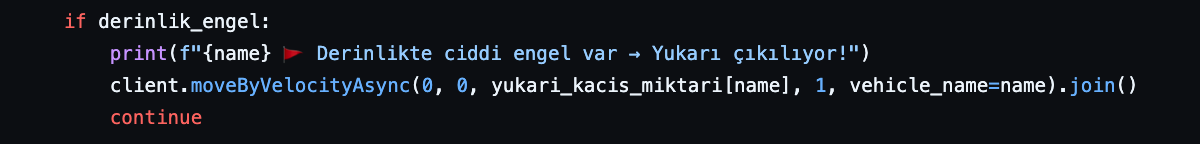
C) Kaya varsa yukarı kaçılır:

  
Şekil.17

Açıklama:

* Kenar tespitiyle çok sayıda sert yapı algılanmışsa (örneğin kaya), drone direkt yukarı çıkar ve turu geçer.

D) Derinlik engeli varsa yukarı kaçılır:

  
Şekil.18

Açıklama:

* Eğer derinlik verisine göre engel çok yakınsa, drone yine yukarı çıkar ve işlem yapılmadan bir sonraki döngüye geçilir.

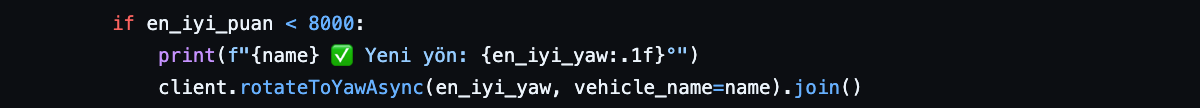
E) Görsel yoğunluk yüksekse en güvenli yön aranır:

  
Şekil.19

Açıklama:

* Eğer orta bölgedeki görsel engel puanı çok yüksekse, uygun\_yon\_bul fonksiyonu çağrılır.
* Bu fonksiyon, çevreyi tarayarak en az engelli açıyı döndürür.

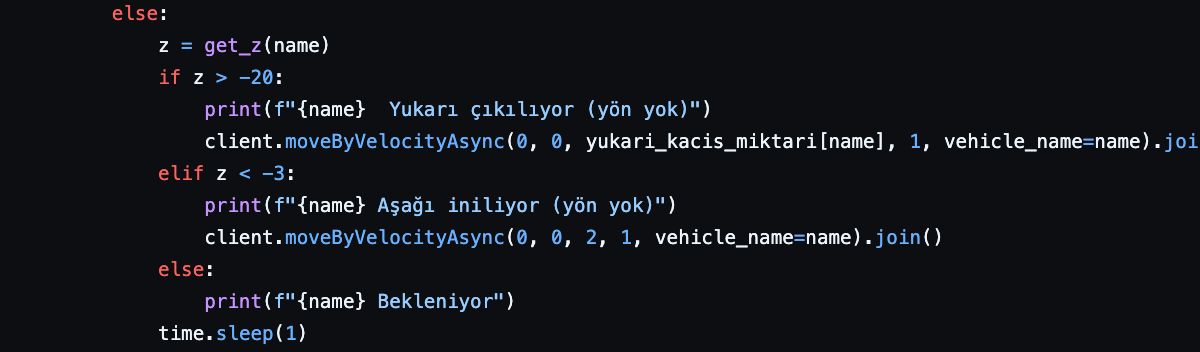
F) Güvenli yön varsa o yöne dönülür:

  
Şekil.20

Açıklama:

* Eğer bulunan yeni yön engel açısından güvenliyse (<8000 puan), drone hemen o yöne döner.

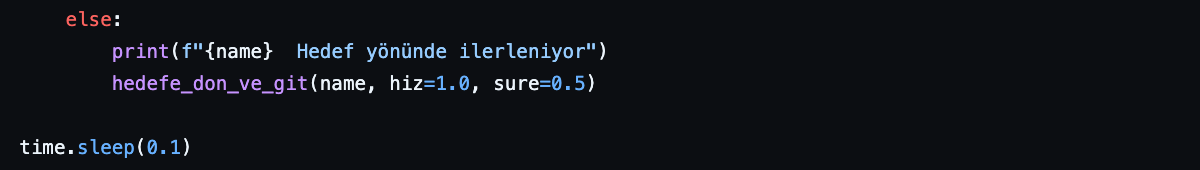
G) Güvenli yön bulunamadıysa yükseklikle oynanır:

  
Şekil.21

Açıklama:

* Eğer hiçbir yön güvenli değilse, yükseklik değiştirilerek yeni bir bakış açısı elde edilmek istenir.
* Yükseklik çok yüksekse aşağı, çok düşükse yukarı çıkılır.

H) Eğer engel yoksa direkt hedefe gidilir:

  
Şekil.22

Açıklama:

* Önünde engel yoksa, drone hedefe doğru döner ve ilerler.

SONUÇ ve ÖNERİLER

Bu çalışmada, **Unreal Engine 4.27** simülasyon ortamında, **Microsoft AirSim** eklentisi ile entegre edilen bir drone’un, **Python diliyle geliştirilen otonom kontrol algoritması** aracılığıyla çevresel engellerden kaçınarak güvenli bir şekilde hedefe ulaşması sağlanmıştır. Sistem, görüntü işleme ve derinlik verisi analizine dayalı olarak çevresel farkındalık elde etmiş, elde edilen bu verilerle dinamik olarak yön ve irtifa kararları alarak otonom uçuş gerçekleştirmiştir.

Drone, ön kamera görüntülerinden ve AirSim tarafından sağlanan derinlik haritalarından faydalanarak; **kaya, ağaç, taş gibi engelleri** tespit etmiş ve bu engellere çarpmadan uygun manevralarla rotasını güncellemiştir.

**Elde Edilen Bulgular**

* Geliştirilen algoritma, drone’un **hedef konuma çarpışmasız şekilde ulaşmasını** sağlamıştır.
* **Derinlik verisi**, çarpışma riski taşıyan nesnelerin tespitinde etkili olmuş ve **gerçek zamanlı manevra kararları** alınmasına katkı sağlamıştır.
* Görsel yoğunluk analizi sayesinde, yalnızca derinlik değil, **görüntü temelli engeller** de (örneğin, renk olarak farklılaşan kayalar) başarılı şekilde algılanmıştır.
* Drone, uçuş boyunca çevre koşullarına göre farklı yönlerden ilerleyerek **en güvenli rotayı dinamik olarak belirlemiştir**.

**Sistemin Güçlü Yönleri**

* Görüntü işleme ve derinlik analiziyle **donanımsız çevre algısı** yüksek başarıyla sağlanmıştır.
* Drone, **tamamen otonom şekilde karar vererek** herhangi bir manuel müdahaleye gerek duymadan görevini tamamlamıştır.
* Geliştirilen yapı, **gerçek dünya uygulamaları için uyarlanabilir parametreler ve algoritmik temeller** sunmaktadır.

**Karşılaşılan Zorluklar**

* AirSim platformunda **derinlik verisinin zaman zaman gecikmeli güncellenmesi**, karar alma sürecinde gecikmelere neden olmuştur.
* Simülasyon ortamındaki **ışık, gölge ve hava koşulları**, görüntü işleme performansını doğrudan etkilemiştir.
* **Küçük boyutlu engeller**, sistem tarafından geç tespit edilmiş ve ani kaçış manevralarına neden olmuştur.

**Gelecek Çalışmalar için Öneriler**

* Sisteme **Lidar** veya **stereo kamera** gibi donanımsal sensörlerin entegre edilmesiyle, daha hassas ve düşük gecikmeli çevresel algılamalar yapılabilir.
* **Derin öğrenme tabanlı nesne tespiti** algoritmaları kullanılarak, engellerin türüne göre farklı stratejiler geliştirilebilir.
* Aynı sistemin **gerçek drone platformunda** test edilerek, simülasyon ile fiziksel dünya arasındaki performans farklılıkları analiz edilebilir.
* **Çoklu drone koordinasyonu** ve **sürü zekâsı algoritmaları** ile daha karmaşık görevlerin eş zamanlı yürütülmesi sağlanabilir.

KAYNAKLAR

[1] Shah, Shital, et al. “AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles.” *Field and Service Robotics*, 2017. Microsoft Research.  
 <https://github.com/microsoft/AirSim>

[2] Bradski, G. “The OpenCV Library.” *Dr. Dobb's Journal of Software Tools*, 2000.

[3] Unreal Engine 4 Documentation. *Epic Games Developer Portal*,  
 <https://docs.unrealengine.com>

[4] Python Software Foundation. “Python Programming Language.”  
 [https://www.python.org](https://www.python.org/)

[5] Gonzalez, Rafael C., and Richard E. Woods. *Digital Image Processing*. Pearson Education, 2018.

[6] Szeliski, Richard. *Computer Vision: Algorithms and Applications*. Springer Science & Business Media, 2010.

ÖZGEÇMİŞ

### **Berk Efe Polat**

2004 yılında doğmuştur. Marmara Üniversitesi Teknik Bilimler Meslek Yüksekokulu Bilgisayar Programcılığı bölümünde öğrenim görmektedir. Yazılım geliştirme, gömülü sistemler ve görüntü işleme alanlarına ilgi duymaktadır. Python, Java ve C# programlama dillerinde deneyime sahiptir.

### **Minel Baştor**

2004 doğumlu Minel Baştor, Marmara Üniversitesi Teknik Bilimler Meslek Yüksekokulu Bilgisayar Programcılığı öğrencisidir. Java, C# ve SQL gibi çeşitli dillerle de uygulamalar geliştirmiştir. Bitirme projesinde drone simülasyonu üzerinde çalışmıştır. Öğrenmeye açık, sorumluluk sahibi ve kendini sürekli geliştirmeyi hedefleyen bir yazılımcı adayıdır.

### **Melike Can**

2005 doğumlu Melike Can, Marmara Üniversitesi MYO Bilgisayar Programcılığı öğrencisidir. Mobil ve oyun geliştirme alanlarına ilgi duymakta; Android Studio, web ve Unity ile projeler geliştirmiştir. Bitirme projesinde Unreal Engine ve AirSim kullanarak bir drone simülasyonu üzerine çalışmıştır. Öğrenmeye açık ve gelişime önem veren bir yazılımcı adayıdır.

### **Serra Çınay Tezer**

2003 yılında doğdu. Marmara Üniversitesi Teknik Bilimler Meslek Yüksekokulu Bilgisayar Programcılığı bölümünde öğrenim görmektedir. Java, Python, C ve C# dillerinde çeşitli projeler geliştirmiştir. Mobil programlama, mobil uygulama geliştirme ve oyun tasarımı konularına ilgi duymakta; bu alanlarda kendini sürekli geliştirmektedir.