

Staj Raporu #1

Barış Ülgen – barisulgen@sabanciuniv.edu

Konu ve Yöntemler

İnsansız hava araçlarının belirlenmiş platformlara otonom inişleri için görüntü temelli bir yöntem konusunu iki ana başlıkta incelemek mümkün. Bunlardan **görüntü işleme** ve **deep learning**'dir.

Görüntü işleme en genel ana hatlarıyla: kamera kalibrasyonu -> görseli yakalama -> thresholding -> feature detection -> bit extraction/marker identification -> pose estimation şeklindedir.

Kamera kalibrasyonu: Kullanılacak kamera ile önceden çekilen referans fotoğraflar yardımı ile kameranın lens karakteristikleri belirlenir, görsel ve gerçek dünya arasındaki “ölçek” belirlenir.

Thresholding: Görsel, bir eşik pixel değeri ile filtrelenip siyah-beyaz (binary) bir görsel oluşturulur.

Feature detection: Çeşitli kenar belirleme (Canny, Sobel, Roberts vb.), köşe belirleme (Harris, Shi/Tomasi vb.) ve en önemlisi de SURF(Speeded up robust features) gibi “özellik” tespit ve analiz eden yöntemlerle görsel analiz edilir.

Marker identification: Daha önceden belirlenmiş etiket sözlükleri ile görsel karşılaştırılarak görseldeki etiketin id numarası tespit edilir.

Pose estimation: Görselin konumu ve kamera arasındaki mesafe ve oryantasyon belirlenir.

Ana kütüphane olarak OpenCV kullanıp python ile çalıştım. Etiket tespit etme ve pose estimation kısımlarında ise bu işlerde kolaylık sağlayan Aruco kütüphanesini kullandım.

En son geldiğim aşamada yukarıdaki basamaklarda pose estimation kısmına kadar geldim. Etiketin mesafesini, x ve y eksenlerini tam anlamıyla tespit edebiliyorken z ekseninde bazı durumlarda tutarsızlık devam ediyor. Bu konudaki denemelerime 2, 3 ve 4 numaralı kaynaklardan ulaşabilirsiniz.

Deep learning kısmı ise aracın çevresini kamera ile yakalayıp, görüntüyü analiz ederek gerekli bilgileri elde ettiği durumda, iniş operasyonunda otonom olarak çalışabilmesi için gereklidir. Asıl hedef; mesafe, hız, rotasyon gibi faktörler ile iniş yapmanın arasındaki korelasyonun belirlenip öğrenilmesidir. Henüz yeni öğrenmekte olduğum, bir data setini yorumlamaya çalıştığım dosyaya 1 numaralı linkten ulaşabilirsiniz. Regression ve classification hakkında bilgi toplayıp bu çalışmada yine python ile devam edip pandas kütüphanesinden yararlandım. İleriki çalışmalarım ise TensorFlow ile devam etmeyi planlıyorum.

Çalışmalar

1) Machine learning test:

<https://www.kaggle.com/barisulgen/redwinerating>

2) Aruco image test:

<https://colab.research.google.com/drive/1ybhl7w4CgG55roSwjPuryM1ZKh4oNj0j>

3) Aruco video test:

File name: "aruco_videoTest2"

4) Camera calibration test (from Tiziano Fiorenzani):

File name: "clb_temp"

Aşağıdaki kaynaklar kısmında yararlandığım ana kaynakları (yararlandığım her kaynağı belirtmenin gereği olmadığından), dosya olarak eklediklerimden ise yararlı bulduğum üç adet makaleyi (paper1, paper2, ve paper3) bulabilirsiniz.

Kaynaklar

- ❖ TensorFlow 2.0 Complete Course - Python Neural Networks for Beginners Tutorial:
<https://www.youtube.com/watch?v=tPYj3fFJGjk>
- ❖ Precision Landing with OPENCV and ARUCO Markers:
https://www.youtube.com/watch?v=wIT_0fhGrGg
- ❖ Kaggle courses:
<https://www.kaggle.com/learn/overview>
- ❖ OpenCV-Python Tutorials
https://docs.opencv.org/master/d6/d00/tutorial_py_root.html
- ❖ Aruco marker basics
https://mecaruco2.readthedocs.io/en/latest/notebooks_rst/Aruco/Aruco.html
- ❖ Geometry of Image Formation:
<https://www.learnopencv.com/geometry-of-image-formation/>
- ❖ Augmented Reality using ArUco Markers in OpenCV (C++ / Python):
<https://www.learnopencv.com/augmented-reality-using-aruco-markers-in-opencv-c-python/>
- ❖ Deep Neural Network Guides Drone to Smooth Landing:
https://www.photonics.com/Articles/Deep_Neural_Network_Guides_Drone_to_Smooth_Landing/a64765

Son olarak, şu an çalışmalarımın image processing kısmını iyi bir şekilde kavramış olmama karşın deep learning kısmında daha çok mesai yapmam gereklidir. Sonraki çalışmalarımda ise ilk hedefim pose estimation kısmındaki sorunların nedenini anlayıp düzeltmek, ikincisi ise deep learning hakkında testlere başlamak olacak.