BİL 467/561 – Image Processing

Summer 2022

Final Project Report

|  |  |
| --- | --- |
| Your Name and Surname: | Harun Serkan Metin |
| Your Student ID: | 201101034 |
| Project Title: | Lane Detection from Stitched Videos |
| Project Personnel: | - |

1. Proje tanımı ve Önbilgi

• Bu projede, arabanın kaput kısmından 3 farklı açıyla kaydedilmiş videoları birleştirerek daha geniş bir görüş elde etmeyi ve bu görüntüden yol şeritlerini tespit etmeyi amaçladım.

• Yol şeritlerini tespit etme sırasında herhangi bir derin öğrenme veya nesne tespit yöntemi kullanmadım. Bu yüzden aldığım çıktılar bazen istenen sonuçtan uzak olabiliyor.

• Soldan, ortadan ve sağdan kaydedilmiş videoları “Stitching” işlemi yaparak birleştiriyorum. Bazı durumlarda şeritler tam olarak algılanabiliyor mesela çok eğimli virajlarda. Bu gibi durumda görüntüyü maskelemek zorlaştığından şerit tespiti için yazdığım kod iyi çalışmayabiliyor.

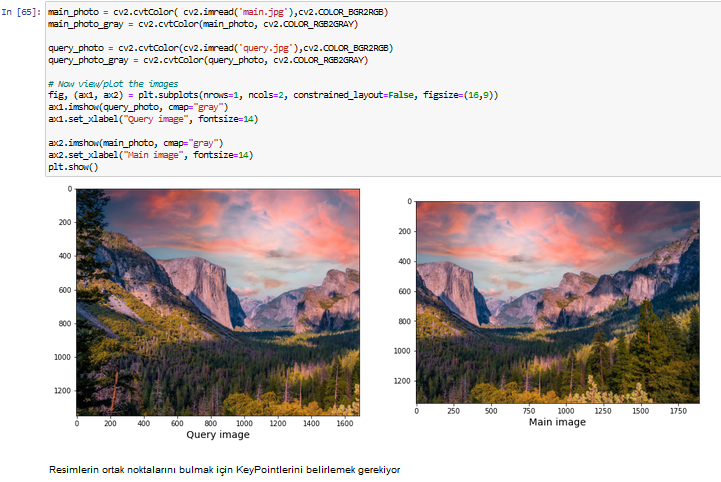
• Her video için KEY POINT (Anahtar nokta) bulmak ve bunları eşleştirmek, “Stitching” için kullanılan bir yöntemdir. Anahtar nokta tespiti için en popüler olan SIFT algoritmasını kullandım. Bu noktaları eşleştirmek için iki farklı fonksiyonu koda implement ettim. İşleyişi uzun zaman alan Brute Force (BF) Metodu ve daha optimal olan K. Nearest Neighbor (KNN) Metodu, kod içinde mod parametresi değiştirilerek kullanılabilir.

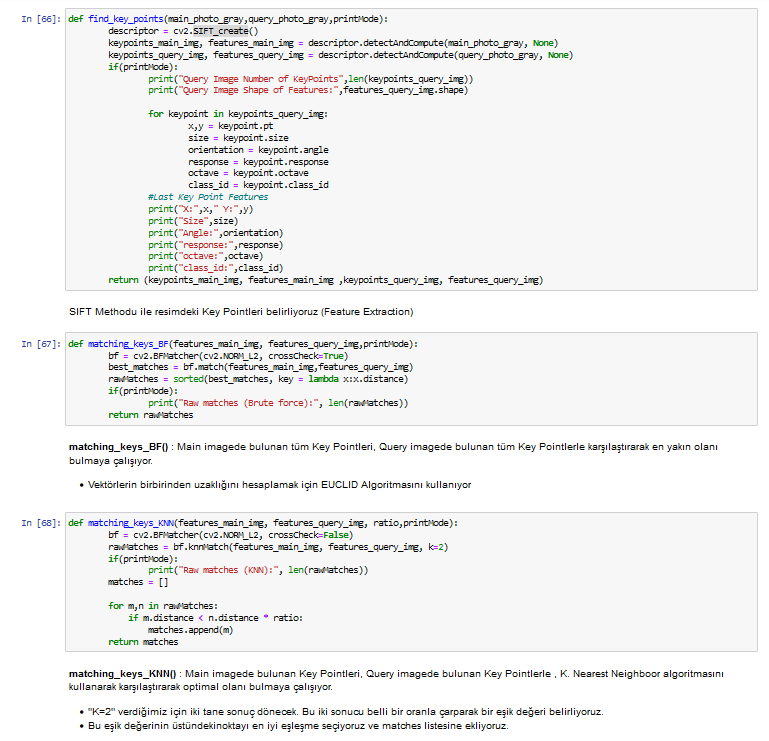
Şerit algılama kısmında, doğru alanı ve bu alandaki doğru şeritleri bulmak için Canny Edge Detection yöntemini ve resmi maskeleme yöntemini kullandım. Maskelemeden sonra beyazla gösterilen kenarları renklendirerek birden fazla çizgi elde ettim. Bu çizgileri iki sınıfta (sol çizgiler, sağ çizgiler) gruplandırdım. Gruplamadan sonra sol çizgileri tek bir çizgide, sağ çizgileri tek bir çizgide birleştirerek 2 şeridi de tespit ettim. Bu şeritlerin arasındaki alanı da Yeşil ile renklendirerek yolun ayırt edilmesini kolaylaştırdım.

1. Tekniksel Çalışma

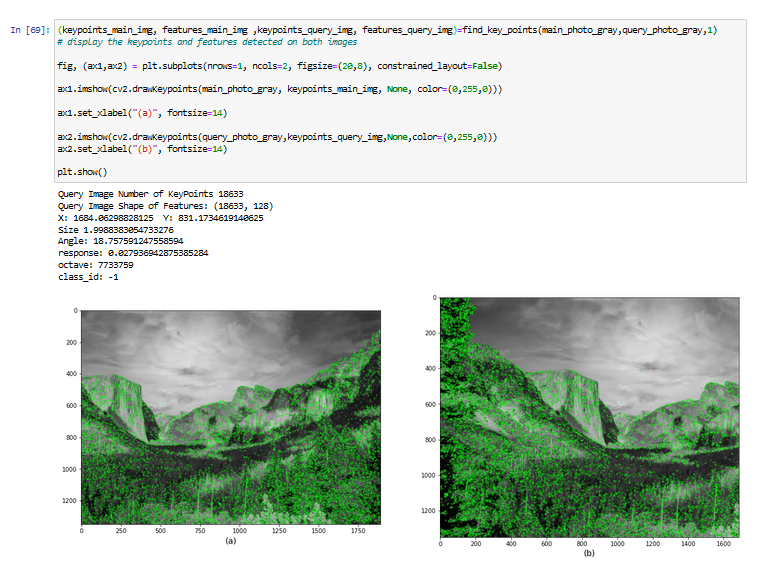
**2.1 Image Stitching:**

Benzer içeriğe sahip farklı resimlerin ortak noktalarını üst üste getirerek eşleşme sağlayacağım (Matching Key Points). Bu eşleme ve birleştirme işlemleri aşağıda gösterdiğim şekilde gerçekleşmektedir.

* Ortak “Key Pointleri” Match edip panorama bir görüntü elde edeceğim resimler.
* Bahsettiğim üzere “Key Pointleri” bulmak ile başladım. Bunun için bir çok method var fakat ben bu projede SIFT methodunu kullandım.
* SIFT şöyle çalışmaktadır:
  + Uzaydaki uç noktaların (minimum-maksimum) elde edilmesi.
  + “Key Pointlerin” konumlarının belirlenmesi.
  + Döngüsel değişime karşı dayanıklılık kazanılması.
  + “Key Pointlerin” tanımlayıcıların bulunması.

Yukardaki görselde belirtilmiş “find\_key\_points()” fonksiyonu SIFT methodunu kullanarak her iki resim için “Key Pointlerini ve Özelliklerini” tuple olarak return ediyor.

Bu noktalardan birbirine benzer olanları bulup, bunları eşleme işlemine Matching denir. Matching için birden fazla yöntem bulunmakta fakat ben bu projede Brute Force (BF) ve K. Nearest Neighbor (KNN) yöntemlerine yer verdim. Kodu çalıştırırken hangi eşleme methodunu kullanacağınıza karar verebilirsiniz.

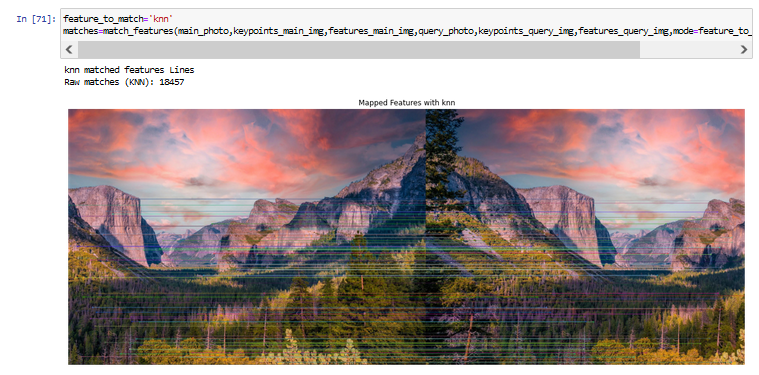


Yukardaki görselde yeşil olarak renklendirilmiş noktalar bu resimlerin SIFT ile bulunmuş Key Pointleridir. Yukardaki kodun çıktısında herhangi bir Key Pointin özellikleri belirtilmiştir.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin içeren bir resim

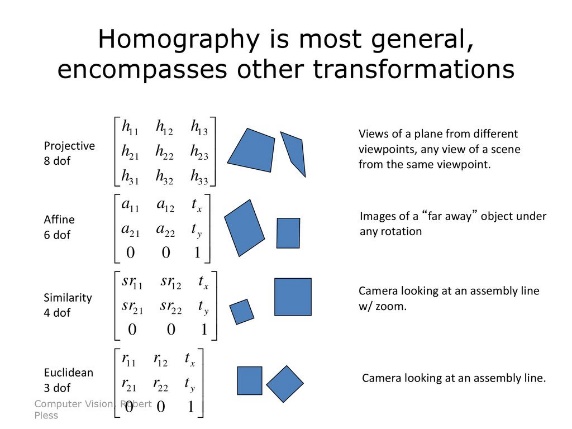
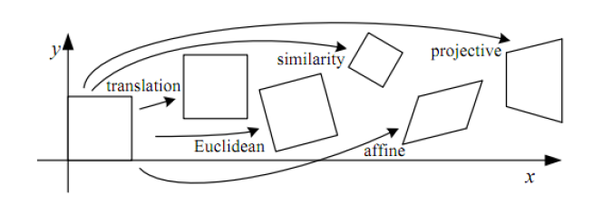
Açıklama otomatik olarak oluşturulduİki resimde de eşleşen noktaların arasına farklı renklerde çizgiler çizerek gösterdim. Eğer resimlerden biri çevrilmemişse yani iki resim de aynı açıdan bakıyorsa bu çizgiler paraleldir. Eğer paralelliği bozan çizgiler varsa yüksek ihtimalle yanlış bir eşleşme olmuştur.

metin, dağ içeren bir resim

Açıklama otomatik olarak oluşturuldu

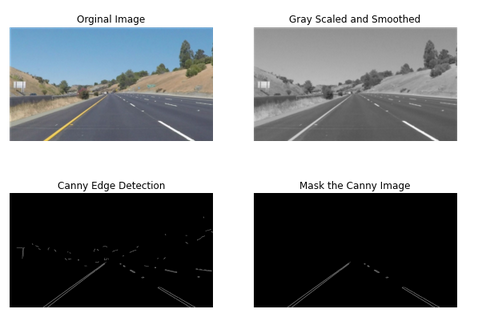
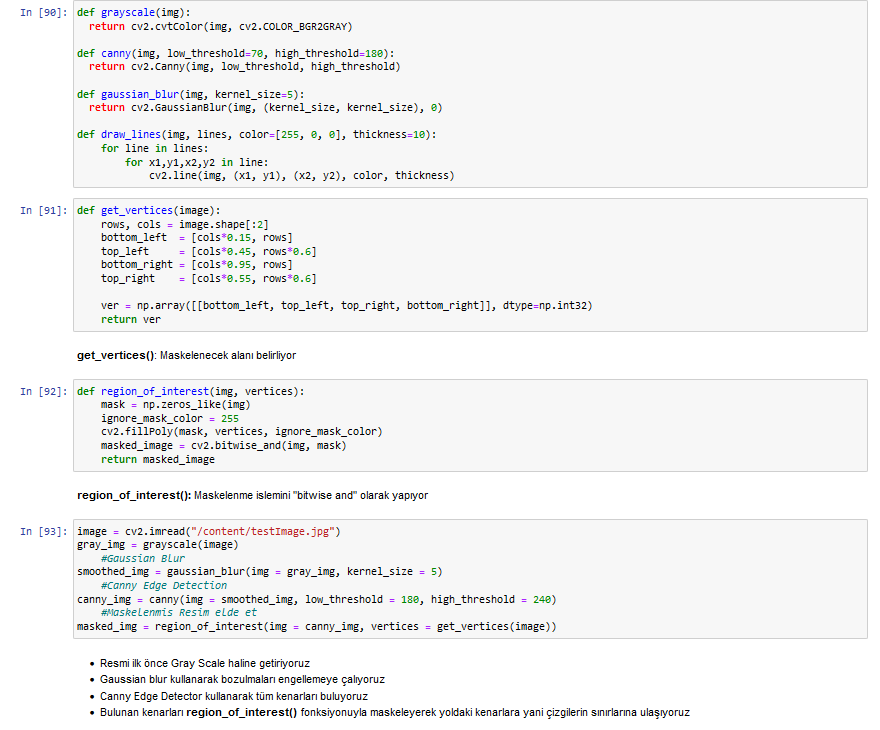
Homography, aynı düzlemsel yüzeyi görüntüleyen iki resmi ilişkilendirmemizi sağlar. Başka bir deyişle farklı açıda veya düzlemde olan resmi, başka bir açıya veya başka bir düzleme taşıma işlemidir.

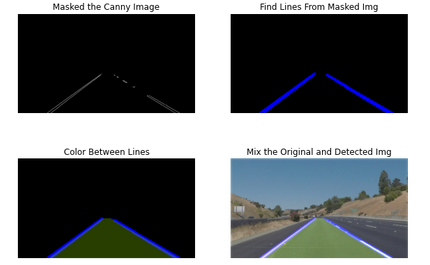
Yukardaki kod parçası, eğer iki resim arasında 4 taneden fazla nokta eşleşmesi varsa bu verilerin Homography Matrisini, RANSAC regresyon kullanarak buluyor. Bulunan Matris ile resim WARP ediliyor ve iki resim üst üste getirilerek eşleşme sağlanıyor.

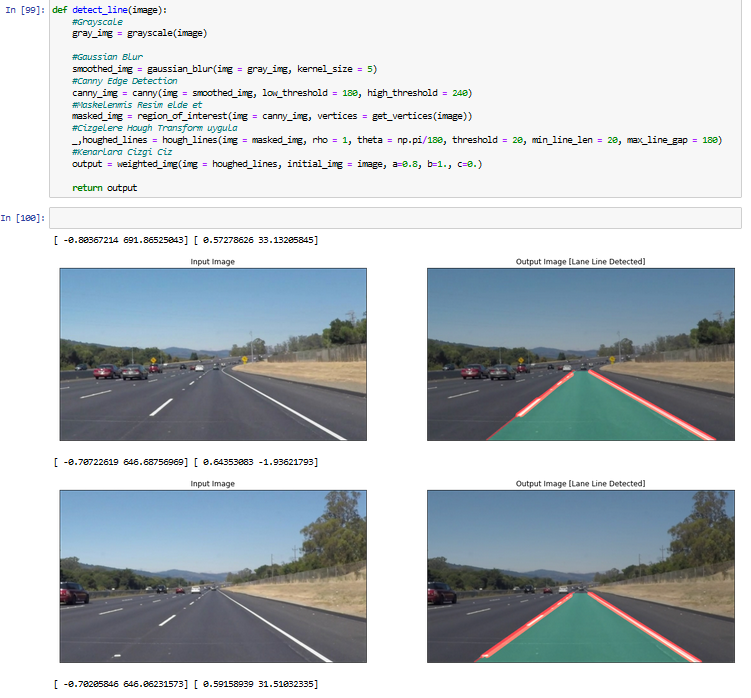
\*RANSAC: **Ra**ndom **Sa**mple **C**onsensus, bir veri kümesinde model parametrelerine bulmak için kullanılan rassal bir yöntemdir. Rastgele bir parametre kümesi oluşturulur ve bu kümeden veriyi en iyi ifade eden parametre seçilir. Böylece veri kümesi modellenmiş olur.

Sonuç olarak aşağıda verildiği gibi bir çıktı elde ediyorum:

**2.1 Lane Detection:**

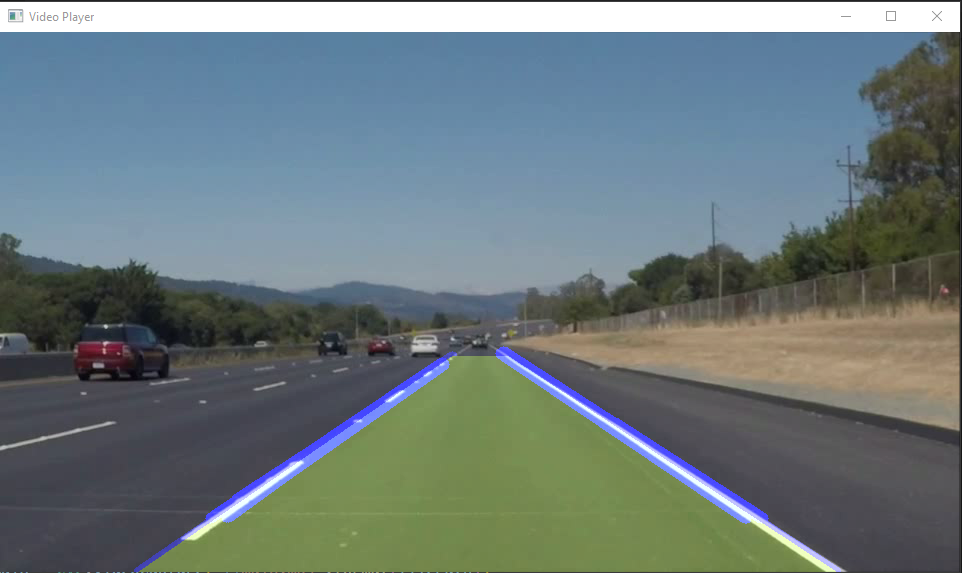
****





1. Bulgular ve Çıktılar

* 2. Kısımda bahsettiğim işlemler fotoğraf kareleri içindi. Ben ise bu işlemleri videolar üzerinde yapmak istediğim için videoların her bir frame’ini alıp bu işlemlerden geçirip başka bir video olarak kaydettim.
* En başta bahsettiğim üzere bir arabanın üzerinden 3 farklı açıdan alınmış görüntüm bulunmamakta. Bu yüzden ben de bir videonun 3 farklı kısmını kırpıp 3 farklı video elde ettim.
* Bu 3 videoyu “Stitch” ettikten sonra Şerit Tespiti yaptım ve elde ettiğim sonuç:



1. Sonuç

Bu ödevlendirmede: SURF, SIFT gibi algoritmaların gerçek hayatta nasıl örneklendiğini, birden fazla metot üzerinde çalışarak Keypointlerin nasıl eşleştirildiğini (BF, KNN, FLANN), homograpy matrisinin ne olduğunu ve warping için ne kadar önemli olduğunu öğrendim.

Ayrıca, uzun soluklu bir projenin devamlılığını nasıl sağlayacağımı, başa çıkmakta zorlandığım problemleri nasıl halledeceğimi, ortaya koyulan çalışmanın her zaman geliştirilebilecek bir noktası olabileceğinin farkına vardım.