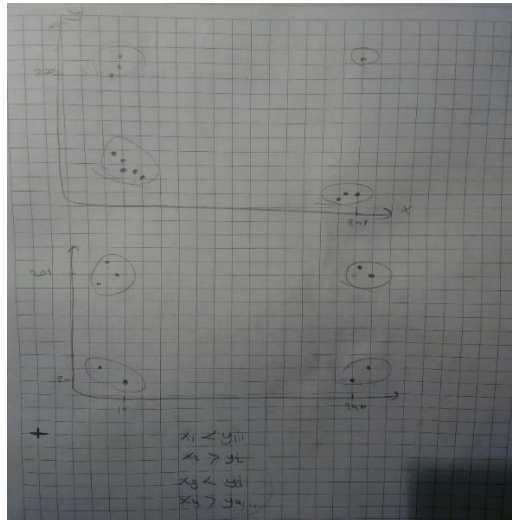


CSE 463 3. ÖDEV RAPORU

ALGORİTMA

İlk başta OpenCV'nin kamera kalibrasyon tahtası kullanılarak kamera kalibrasyonu yapıldı ve kamera matrisi ile distorsion katsayılarını elde edildi. Bunun için calibrate kamera örnek kodunu kullanıldı. Çizilmek istenen kübün 3 boyutlu eksenlerdeki koordinatlarını vektöre kaydedildi. 2. Ödevdeki gibi kameradan sürekli görüntü alınarak aşağıdaki işlemler gerçekleştirildi.

- Kameradan bir görüntü al.
- Alınan görüntüyü grayscale'e çevir.
- Görüntüyü binary resme çevir.
- Elde edilen görüntü üzerinde Canny edge detektörünü çalıştır.
- Görüntü üzerindeki çevreleri(contour) ve bu çevreler arasındaki hiyerarşiyi findContours fonksiyonu ile bul.
- Bulunan contourları ve hiyerarşiyi kullanarak görüntü üzerinde çevreleri dolaşarak köşe olabilecek noktaları bul.
 - Bunun için gelen değerleri inceledim ve bir şeklin dikdörtgen olması için gelen değerlerin ortalamasından büyük ve küçük değerlerin gelmesi gerektiğini farkettim; yani (x_{kucuk}, y_{kucuk}) (x_{kucuk}, y_{buyuk}) (x_{buyuk}, y_{kucuk}) (x_{buyuk}, y_{buyuk}) değer ikililerin gelmesi gerektiğini farkettim



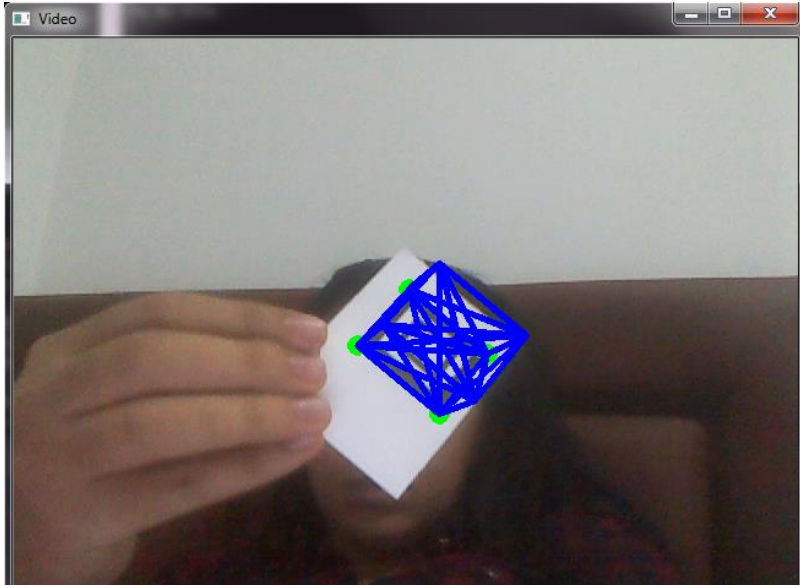
- Her çevre noktasının değerlerini topladım ve ortalamalarını aldım.
- Yukarıda bahsettiğim büyük ve küçük değerlerinin sınırını ortalama olarak belirledim
 - Yani eğer x değeri ortalamadan küçük ise x_{kucuk} tür ortalamadan büyük ise x_{buyuk} tür

- Hiyerarşi kullanımından bahsedecek olursak, Bir çevrenin çocuk sayısı bitene kadar şekil içinde dolaştım. Çocuk sayısı bitimi contour arrayinde -1 olarak ifade edilmiştir.
 - Şekilin Dikdörtgen olup olmadığına baktım. Bunun için yukarıda bahsettiğim yöntemi kullandım.
 - Eğer nokta değerleri bahsettiğim şekilde mevcut ise bu çevre dikdörtgendir.
 - Hata: Açık kontrolü yerine nokta kontrolü yaptığım için şekil de dikdörtgen baklava dilimi şeklinde gelse bile onu dikdörtgen olarak kabul ediyor.
- Bulunan şekilleri gösterebilmek için kameradan gelen görüntünün clone'unu alarak yeni bir görüntü elde ettim
- Dikdörtgenleri bulduktan sonra onun içindeki çocuklara baktım.
 - Eğer çocuk yoksa bu birinci fiducialdır
 - Tek çocuğu varsa bu ikinci fiducialdır
 - İki çocuğu varsa bu üçüncü
- Herhangi bir tane fiducial bulunduğunda fiducial arama kısmından çıktım
- Bulunan fiducialın içindeki köşe noktalarını imagePoint olarak sakladım
- Daha önce Kameranın içsel parametrelerini ve distortion değerlerini bulmuştum. Noktaları 3 boyutlu dünyadan 2 boyutlu dünyaya taşımak için Rotation ve Translation matrisine de ihtiyacım vardı. Bunun için openCv solvePnp fonksiyonunu kullandım
- Elde edilen matrisler yardımı ile en başta belirlenen 3 boyutlu dünyada çizmek istediğim küp noktalarının 2 boyutlu dünyadaki karşılığını opencv nin projectPoints fonksiyonu ile elde ettim.
- 2 boyutlu dünyadaki noktaların her birinin arasına line çizerek küp şeklini elde ettim.
- Küp şeklini var olan frame e ekleyerek kullanıcıya gösterdim.

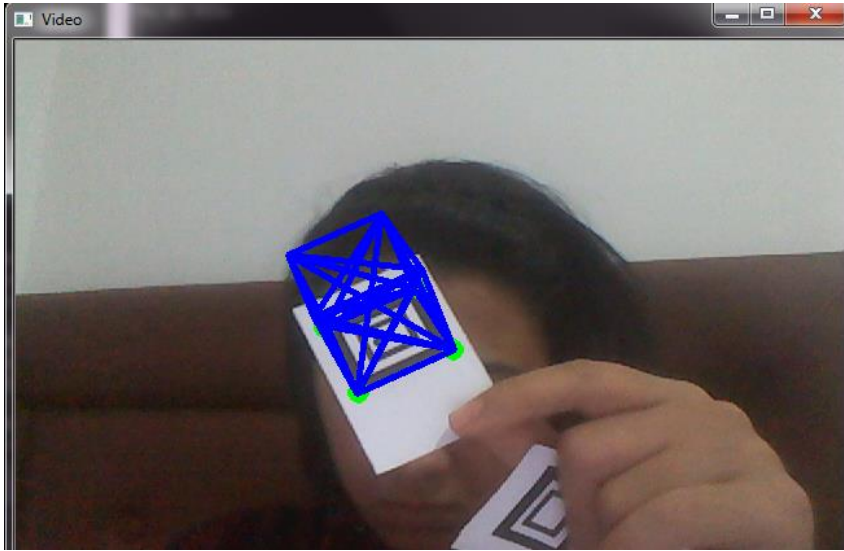
Programın Çalışma Örnekleri



İki noktalı marker



Noktasız marker



Tek noktalı marker